Multi-Object Tracking Via Multi-Attention

1nd Xianrui Wang
dept. Computer Science and Technology
Huazhong University of Science and Technology
Wuhan, China
wxrui@hust.edu.cn

2st Hefei Ling
dept. Computer Science and Technology
Huazhong University of Science and Technology
Wuhan, China
lhefei@hust.edu.cn

3rd Jiazhong Chen
dept. Computer Science and Technology
Huazhong University of Science and Technology
Wuhan, China
jzchen@hust.edu.cn

4th Ping Li
dept. Computer Science and Technology
Huazhong University of Science and Technology
Wuhan, China
lpshome@hust.edu.cn

***Abstract*—Data association plays a crucial role in Multi-Object Tracking(MOT), but it is usually suppressed by occlusion. In this paper, we propose an online MOT approach via multiple attention mechanism(Multi-Attention) to handle the frequent interactions between targets. Specifically, the proposed Multi-Attention consists of spatial-attention, channel-attention, and temporal-attention three modules. The spatial-attention module lets the network focus on visible local areas by generating a visibility map, and the channel-attention module combines texture information and context information adaptively to build a recognizable object descriptor, then the temporal-attention module pays different attention to objects in the same trajectory avoiding the suppress caused by contaminated samples. Besides, a multiple branch convolutional block called receptive filed module(RFModule) is introduced to learn multiple levels of information for Multi-Attention. The experimental results on MOTChallenging benchmarks demonstrate the effectiveness of the proposed MOT algorithm against both online and offline trackers.**

***Index Terms*—Multi-object tracking, Attention mechanism, Receptive filed, Data association**

I. Introduction

Object tracking is one of the most challenging tasks in computer vision, which plays an important role in autonomous driving, video surveillance, etc. It can be divided into Single Object Tracking(SOT) and Multiple Object Tracking(MOT). The former locates and associates only one object in video frames, and the latter should identify multiple objects focusing on the distinction between different objects. And MOT can be further categorized into online and offline methods. Offline methods use all the video frames information(both past and future) to generate trajectories while online methods use only past frames information. Although offline methods have excellent performance, online methods are more applicable to real-time applications.

Since significant improvement has been achieved on object detection problems, tracking-by-detection has become an effective method to get better performance on MOT. At first, all objects going to be tracked are located in each frame by an object detector. Then, the same detections across different frames will be linked by data association algorithms. And the data association algorithm usually consists of three steps: feature extraction for object representation, similarity calculation to measure each association and object assignment to find the optimal association. In general, the performance of data association heavily relies on the similarity score based on the feature extracted from the detected object. Most existing approaches take object sequence as input and directly extract appearance feature from the cropped object patches, the performance will be limited by many factors. First, when there are multiple objects in the same video frame, every object should be sent to the network separately, this undoubtedly increases the cost of time during inference. Second, object sequences only include the foreground information, while the context information outside the foreground is also helpful for identifying objects, especially for objects with low visibility as shown in Fig. 1. Furthermore, when objects are obscured caused by interaction among objects, the noisy features extracted from these objects will also update the model less effectively.

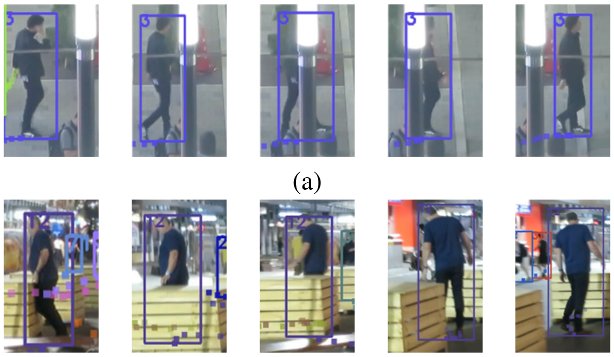

(a)
(b)

Fig. 1. Some examples in MOT17 dataset. The easy examples can be identified with global texture information, while local texture information and global context information may be helpful for occlusion.

These factors require us to design an appearance model to extract robust feature representation for effective data associ-

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

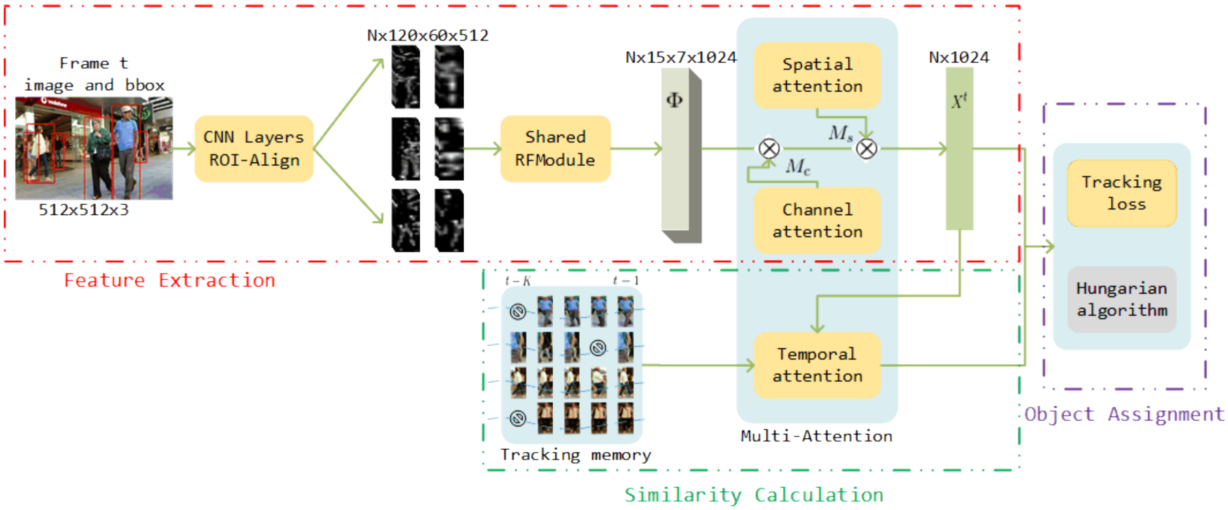


Fig. 2. Architecture of the proposed method. The Network consists of feature extraction, similarity calculation, and object assignment. For feature extraction, a shared RFModule is applied to whole objects to extract multi-level appearance features Φ for individual objects. And spatial-attention and channel-attention module generate recognizable object descriptors as X^t by focusing on visible local regions. For similarity calculation, temporal-attention allocates different importance to objects in trajectories to get better performance. For object assignment, we use the Hungarian algorithm to get the optimal assignment during inference, while using tracking loss to update the network during training.

ation. To address the issues observed, we make the following contributions in this work:

- Instead of object sequence, we use image sequence as the network input and then extract individual feature for each object using ROI-Align [29]. In this way, the network can learn both texture information and context information from training data, which is helpful to handle contaminated examples. Correspondingly, the network output similarity matrix to measure the affinity between objects and trajectories in parallel rather than a single similarity score, and it can reduce the time complexity.
- We propose a multiple branch convolutional block called Receptive Filed Module(RFModule) to enhance the discriminability and robustness for the appearance features extracted from CNN layers. Therefore, the proposed network can learn multi-level features combining fine-grained texture feature and coarse-grained context feature which is helpful to identify objects with low visibility.
- Since the proposed network has learned multi-level features, we adopt the Multi-Attention network to generate adaptive descriptors for different objects automatically. First, the Spatial-Attention lets the network focus on visible local regions. Then, the Channel-Attention allocates different importance to the fine-grained features and coarse-grained features to make up a recognizable object descriptor. Finally, the Temporal-Attention adaptively pays different attention to different objects in the same trajectory, to avoid being suppressed by contaminated objects.

II. Related Work

The researches on MOT can be categorized into detection based tracking and detection free tracking depending on whether detections are given or not.

A. Detection Based Tracking

With the development of object detection, detection based tracking has become a popular strategy in MOT. At first, a pre-defined object detector is applied to each frame to get all of the object locations. Then, MOT algorithm associates object detection results across video frames to generate trajectories, and this step is usually called data association. [1] [2] proposed standard benchmarks for pedestrians tracking. According to whether future information used or not, MOT methods can be further split into online and offline methods.

1) Offline Methods: The offline methods usually consider the data association problem as a global optimization problem and focus on designing various optimization algorithms. Some methods cast data association into network flow problem, [3] solved a constrained flow optimization problem for multiple object tracking, and used k-shortest paths algorithm for associating the tracks, [4] added a pairwise cost to the min-cost network flow framework and proposed a convex relaxation solution with a rounding heuristic for tracking. And these are also some methods try to solve it with minimum subgraph cut algorithm, [6] select hypotheses spatially and link these over time while maintaining disjoint path constraints, and evidence about pairs of detection hypotheses is incorporated whether the detections are in the same frame, neighboring frames or distant frames. [7] introduce a novel local pairwise feature based on local appearance matching that is robust to partial occlusion and camera motion and employ an efficient primal feasible optimization algorithm to the subgraph multi-cut problem.

2) Online Methods: On the other hand, data association is usually treated as the multidimensional assignment(MDA) problem under online methods and then solved with Hungarian or Munkres algorithm. These methods focus on improving the measurement of object similarity. In the early stage, some hand-crafted features such as HOG are used, and recently, more and more deep learning networks are proposed. [8] proposed the spatiotemporal context learning algorithm to build a robust appearance model, [10] proposed a siamese network to encode both appearance information from RGB images and the motion information from the optical-flow map, [12] combined multi-scale discriminative feature and spatiotemporal motion feature to enhance the performance of MOT. And in order to build an end-to-end network, [14] modified Hungarian algorithms to obtain a differentiable framework. There are also some methods build a relation network that is popular in Natural Language Processing(NLP), combining appearance, motion, location cues together over a long period of time to strengthen feature representation, such as [15] [16]. Our proposed method also follows online tracking within detection based tracking paradigm, and mainly focuses on strengthening the discriminability and robustness of appearance feature to improve the measurement of object similarity.

B. Detection Free Tracking

Since significant progress has been made on single object tracking in recent years, some works apply the state-of-the-art single object tracker in MOT directly. [18] partitions the state space of the target into four subspaces and only utilizes single object trackers to track targets in tracked state. [20] use target specific classifiers to compute the similarity for data association in a particle filtering framework. [21] keep both the tracking results of single object trackers and the object detections as association candidates and select the optimal candidate using an ensemble framework. The detection free tracking methods get rid of dependence on detection, but even with a proper target management mechanism, directly applying multiple SOT trackers simultaneously to track multiple targets still experiences various difficulties. Firstly, the MOT should create or destroy a tracker automatically and appropriately which is not required in SOT. Secondly, the single object tracker aims to track one object and separates it from the background, so it is prone to drift due to frequent interactions between different objects. To address these problems, [14] build an end-to-end trainable single object tracker for MOT, [19] proposed a cost-sensitive tracking loss to limit the drift case. [22] proposed an instance-aware tracker to integrate SOT techniques for MOT by encoding awareness both within and between target models.

III. Problem Formulation

Following the detection based tracking paradigm, we formulate the MOT problem as an MDA problem. At frame t, the existing trajectories can be represented by $T = \{T_i\}_{i=1}^{M_t}$, where M_t denotes the number of trajectories. And each trajectory can be represented by a series of objects

$$T_i = \{O_i^\tau\}_{\tau=t-K}^{t} \tag{1}$$

the $O_i^\tau = [x_i^\tau, y_i^\tau, w_i^\tau, h_i^\tau]$ indicates same obejct's location in recent K frames. x_i^τ and y_i^τ denote the center of object i, and w_i^τ and h_i^τ denote the width and height of object respectively. And in addition, the set of object locations at frame t are represented as $O^t = \{O_i^t\}_{i=1}^{N_t}$, N_t is the number of objects. Our goal is to calculate similarity between objects and trajectories, and it can be split into two steps.

1) similarity between objects: Given two frames I^{t_1} and I^{t_2}, the set of objects can be represented by $O^{t_1} = \{O_i^{t_1}\}_{i=1}^{N_{t_1}}$ and $O^{t_2} = \{O_i^{t_2}\}_{i=1}^{N_{t_2}}$, and we can obtain the similarity matrix as following:

$$P^{t_1 t_2} = \{sim(O_i^{t_1}, O_i^{t_2})\}, i = 1 \ldots N_{t_1}, j = 1 \ldots N_{t_2} \tag{2}$$

the $sim(O_i^{t_1}, O_i^{t_2})$ indicates the similarity between $O_i^{t_1}$ and $O_j^{t_2}$ in different frame, and the detail will be introduced in Sec. IV-C3.

2) similarity between objects and trajectories: In MOT problem, we need calculate the similarity between objects in current frame and existing trajectories at frame t, and it can be constituted by the linear combination of similarity between objects:

$$\begin{gathered} S^t = \{\sum_{\tau=t-K}^{t} \alpha_{i\tau} sim(O_i^\tau, O_j^t)\} \\ O_i^\tau \in T_i, i = 1 \ldots M_t, j = 1 \ldots N_t \end{gathered} \tag{3}$$

and the M_t indicates the number of existing trajectories, the N_t indicates the number of objects in current frame I^t, the weight coefficient $\alpha_{i\tau}$ is calculated by temporal attention which will be introduced in Sec. IV-C3.

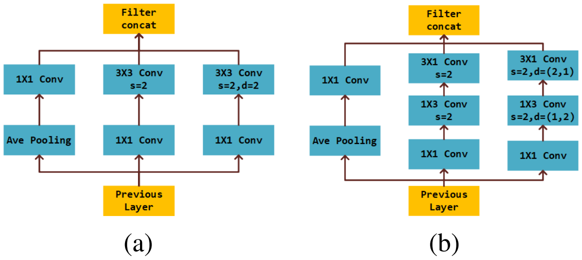
IV. Proposed Approach

A. Architecture Overview

The Overview of the proposed algorithm is shown in Fig.2. At frame t, the network has saved K previous frames information in tracking memory, then it gets current frame I^t and object detection O^t as input. For feature extraction, we use the first two layers of ResNet-50 proposed in [28] and ROI-Align proposed in [29] to extract individual features for each object. The RFModule is introduced to enhance the discriminability and robustness of appearance features by learning multi-level features. Spatial-attention and channel-attention are applied to generate a recognized object descriptor. For similarity calculation, we allocate different importance to different objects in the same trajectory by using temporal-attention, in order to avoid the limitation caused by occlusion. For object assignment, the Hungarian algorithm is applied to the similarity matrix getting the optimal association.

B. RFModule

The proposed RFModule shown in Fig.3 is a multi-branch convolutional block. In object detection and classification tasks, it is a simple and natural way to apply different kernels to learn features with multi-size receptive filed in CNNs, which is supposed to be superior to the fixed size, and we try to introduce this mechanism to object tracking. Inspired



(a) (b)

Fig. 3. The architecture of RFModule. (a) shows the original architecture modified from Inception. (b) shows the factorization of the $n \times n$ convolutions in order to reduce parameters and calculation.

by the Inception-V4 network [30], we modified the bottleneck structure to strengthen the discriminability and robustness of appearance feature to improve the measurement of object similarity. As Fig.3 shows, first, we employ a 1×1 conv-layer at each branch to reduce the number of channels in the feature and 3×3 and 5×5 conv-layer are added to branch2 and branch3 separately, but replace 5×5 conv-layer by 3×3 conv-layer with dilation 2, considering they both have same receptive filed and latter has fewer parameters. finally, to reduce parameters and computation, we use 1×3 plus an 3×1 conv-layer to replace the original 3×3 conv-layer.

C. Multi-Attention

It is well known that attention plays an important role in human perception, and recently, there have been several attempts to incorporate attention processing to improve the performance of CNNs in the classification and object detection tasks. [25] used an encoder-decoder style attention by refining the feature maps to get a better performance in the classification task, [26] proposed the SEBlock to achieve channel attention. And [27] applied spatial attention and channel attention in the object detection task. Inspired by these works, we introduce the Multi-Attention mechanism which consists of three modules to multiple object tracking task.

1) channel-attention: We build the channel attention module using the inter-channel relationship of features to select appropriate descriptors for different objects. As each channel of a feature map is considered as an object descriptor, the channel attention is used to select which is more powerful to identify the object. We denote the feature map extracted from the last convolutional layer of RFModule as $\Phi \in \mathbb{R}^{H \times W \times C}$. First, both the average-pooling and max-pooling operations are applied to Φ, and we can get two different descriptors $F_{avg}^c \in \mathbb{R}^{1 \times 1 \times C}$ and $F_{max}^c \in \mathbb{R}^{1 \times 1 \times C}$. Then, the sum of the descriptors are forwarded to a point-wise convolutional layer whose kernel size is 1×1. Finally, the $sigmoid$ function is applied to calculate the normalized attention score. In short, this module can be represented as:

$$M_c(\Phi) = \sigma(f_{1 \times 1}(F_{max}^c + F_{ave}^c)) \tag{4}$$

where σ denotes the sigmoid function and $f_{1 \times 1}$ denotes the convolutional layer with kernel size of 1×1, F_{max}^c and F_{ave}^c denotes the pooling operation along the spatial axis. As we have get channel attention map, the final features can be calculated as:

$$\Phi_c = M_c(\Phi) \otimes \Phi \tag{5}$$

where $\otimes$ denotes a broadcasted element-wise multiplication.

2) spatial-attention: We build the spatial attention module using the inter-spatial relationship of features to focus on unobstructed local areas. The spatial-attention module is followed after the channel-attention module, and get Φ_c as input. First, we apply the average-pooling and max-pooling operations along the channel axis to get different descriptors $F_{ave}^s \in \mathbb{R}^{H \times W \times 1}$ and $F_{max}^s \in \mathbb{R}^{H \times W \times 1}$. Then, a convolutional layer with 7×7 kernel is applied to extract position information from the sum of two descriptors. Finally, we use the $sigmoid$ function to calculate the normalized attention score. In short, this module can be represented as:

$$M_s(\Phi_c) = \sigma(f_{7 \times 7}(F_{ave}^s + F_{max}^s)) \tag{6}$$

where σ denotes the sigmoid function and $f_{7 \times 7}$ denotes the convolutional layer with kernel size of 7×7. As we have get the spatial attention map, the final features can be calculated as:

$$\Phi_s = M_s(\Phi_c) \otimes \Phi_c \tag{7}$$

where $\otimes$ denotes a broadcasted element-wise multiplication. After all, an average-pooling is applied to Φ_s, and we get the final feature $X \in \mathbb{R}^C$ strengthened by attention mechanism to describe the object.

3) temporal-attention: Since our existing trajectories often contain contaminated objects, such as occlusion or misalignment, simply assigning equal weight to all the objects in the same trajectory may degrade the model performance. To avoid the effect of contaminated objects, we exploit the temporal-attention module to adaptively allocate different attention to different objects in the trajectory. The attention scores between the current objects and objects in existing trajectories are inferred from the introduced spatial-attention and channel-attention:

$$\alpha_\tau = a_1 M_c^\tau \cdot M_c^t + a_2 M_s^\tau \cdot M_s^t + a_3, \tau = t - K \dots t \tag{8}$$

where α_τ denotes the weight coefficient between the current object and one of the existing trajectories at frame t, $M_s = flat(M_s(\Phi_c)), M_s \in \mathbb{R}^N, N = H \times W$ is reshaped from spatial attention map, and $M_c = flat(M_c(\Phi)), M_c \in \mathbb{R}^C$ is reshaped from channel attention map, a_1, a_2, a_3 are learnable parameters.

Now we can calculate the weight coefficients α_τ in (3) using temporal attention module, and we measure the similarity between different objects using the extracted object descriptor:

$$sim(O_i^{t_1}, O_j^{t_2}) = X_i^{t_1} \cdot X_j^{t_2} \tag{9}$$

So, we can get the similarity matrix between objects at current frame and existing trajectories following (3).

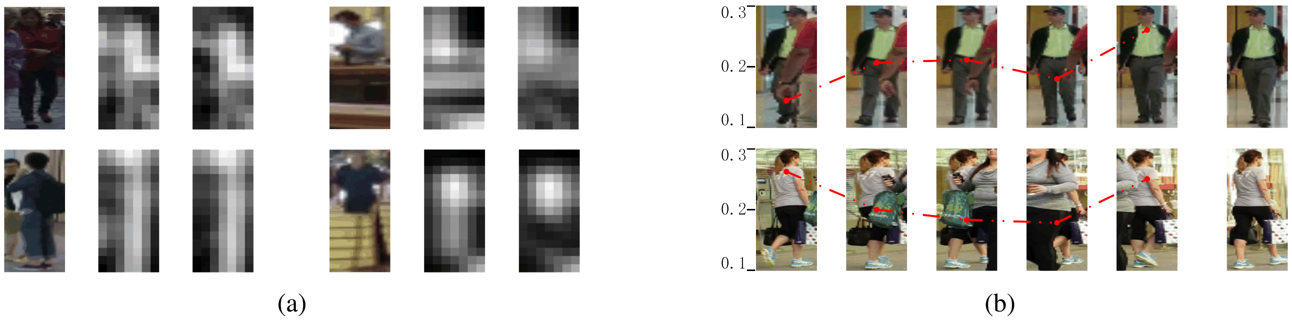


Fig. 4. Visualization of the Multi-Attention. (a) We list some examples from MOT17 dataset, for each group, we can see the original image patch, features before and after Multi-Attention. And the visible local regions have a higher activation value in the proposed network. (b) shows the weight scores calculated from temporal attention, the occlusion get a lower importance while calculating similarity score.

D. Tracking Loss

First of all, we formulated the target matrix G from the ground truth

$$G_{ij} = \begin{cases} 1, O_j \in T_i \\ 0, O_j \notin T_i \end{cases}, i = 1 \dots M_t, j = 1 \dots N_t \tag{10}$$

If none of the objects belongs to T_i, $\sum G_{i:} = 0$, otherwise $\sum G_{i:} = 1$, and correspondingly, if O_j don't belong to any trajectory, $\sum G_{:j} = 0$, otherwise $\sum G_{:j} = 1$.

As mentioned in the previous section, we finally get the similarity matrix S between objects and existing tracking trajectories, S_{ij} indicates the similarity between i^{th} trajectory and j^{th} object. On the one hand, we apply a row-wise softmax operation on the matrix to get the probability distribution for every trajectory as $S1$, on the other hand, a column-wise softmax operation is applied to get the probability distribution for every object as $S2$. During training we optimize the following multi-part loss function:

$$\begin{aligned} loss = & \sum_{i \in \Omega_1^m} \sum_{j=1}^{N_t} G_{ij} focalloss(S1_{ij}) \\ & + \sum_{i \in \Omega_1^{um}} \sum_{j=1}^{N_t} (S_{ij} > \sigma) focalloss(1 - S1_{ij}) \\ & + \sum_{i=1}^{N} \sum_{j \in \Omega_2^m} G_{ij} focalloss(S2_{ij}) \\ & + \sum_{i=1}^{N} \sum_{j \in \Omega_2^{um}} (S_{ij} > \sigma) focalloss(1 - S2_{ij}) \end{aligned} \tag{11}$$

where Ω_1^m and Ω_1^{um} denote if there is a object belongs to trajectory T_i, and correspondingly Ω_2^m and Ω_2^{um} denote if the object O_j belongs to any trajectory. Specially, for the mismatched case, we only penalizes that the predicted similarity score is greater than a predefined constant σ. In order to focus learning on hard examples, we also use the $focalloss$ proposed in [23].

$$focalloss(p_t) = -\alpha(1 - p_t)^\gamma \log(p_t) \tag{12}$$

E. Motion Model

During inference, a simple motion model following [19] was built to select candidate objects before calculate appearance similarity. We formulate the center of object as a linear motion with constant velocity, and keep the scale of object unchanged. Denote $C_i^t = [x_i^t, y_i^t]$ and V_i^t as the center and velocity of object i as frame t, the predicted location can be defined as

$$C_i^{t+1} = C_i^t + V_i^t \tag{13}$$

and the velocity can be represented by

$$V_i^t = \frac{1}{K}(C_i^t - C_i^{t-K}) \tag{14}$$

the K is same as the one in (1), indicating the length of trajectories saved in our proposed network. Given the predicted location for the next frame, we only consider detections surrouding the predicted location as candidate objects and calculate their similarity score.

V. Experiments

In this section, we present the experimental results on public avaliable MOTChallenge benchmark and analysis for the proposed online MOT algorithm.

A. datasets

The MOTChallenge benchmarks are widely used in the field of multi-object tracking to evaluate different trackers. The MOT16 dataset consists of 7 training sequences and 7 testing sequences, including indoor and outdoor scenes of public places with crowded pedestrians as the objects of interest. Both training and testing sequences provide pedestrian detections, but manually annotated ground-truth bounding boxes are only provided for training sequences. While The MOT17 dataset consists the same sequences as MOT16, it provides three sets of different detections(DPM [31], Faster-RCNN [32], and SDP [33]) additionally for more comprehensive evaluation.

B. evaluation metrics

To evaluate the performance of proposed MOT method, we consider the standard following metrics used by MOT benchmarks:

TABLE I
TRACKING PERFORMANCE ON THE MOT17 DATASET.

Mode	Method	**MOTA**(↑)	**MOTP**(↑)	**IDF1**(↑)	**MT**(↑)	**ML**(↓)	**FP**(↓)	**FN**(↓)	**IDSw**(↓)
Offline	IOU [9]	45.5	76.9	39.4	15.7%	40.5%	**19,993**	281,643	5,988
	MHT_bLSTM [13]	47.5	77.5	**51.9**	18.2%	41.7%	25,981	268,042	**2,069**
	EDMT [5]	50.0	77.3	51.3	**21.6%**	**36.3%**	32,279	**247,297**	2,264
	MHT_DAM [11]	**50.7**	**77.5**	47.2	20.8%	36.9%	22,875	252,889	2,314
Online	FPSN [12]	44.9	76.6	48.4	16.5%	**35.8%**	33,757	269,952	7,136
	DMAN [19]	48.2	75.7	**55.7**	**19.3%**	38.3%	26,218	263,608	**2,194**
	E2EM	47.5	76.5	48.8	16.5%	37.5%	**20,655**	272,187	3,632
	AM_ADM17 [17]	48.1	76.7	52.1	13.4%	39.7%	25,061	265,495	2,214
	Ours	**48.5**	**77.0**	44.9	17.7%	38.6%	25,739	**261,710**	3,089

- MOTA: Multiple Object Tracking Accuracy which combines three error sources: false positives, missed targets and identity switches, is the main metrics to evaluate MOT performance.

$$MOTA = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t} \qquad (15)$$

  where m_t, fp_t, and mme_t are the number of misses, of false positives, and of mismatches, respectively, for time t.
- MOTP: Multiple Object Precision which measures the misalignment between the annotated and the predicted bounding boxes.

$$MOTP = \frac{\sum_{it} d_t^i}{\sum_t c_t} \qquad (16)$$

  It is the total error in estimated position for matched object-hypothesis pairs over all frames, averaged by the total number of matches made.
- IDF1: the ratio of correctly identified detections over the average number of ground-truth and computed detections
- MT: the ratio of ground-truth trajectories covered by an output trajectory for at least 80% of ground truth length
- ML: the ratio of ground-truth trajectories covered by an output trajectory for at most 20% of ground truth length
- FP: the number of False Positives
- FN: the number of False Negatives
- IDSw: the number of Identity Switch

C. implement details

The proposed method is implemented using the popular deep learning framework PyTorch and runs on a desktop with 4 Titan Xp GPU. We adapt the first two layers of ResNet-50 as our backbone network, and modify the rest layers with RFModule after ROI-Align to extract robust features. Given the current frame at time t, we resize it to the resolution of 512×512 after data augmentation, and the resolution of features after ROI-Align is $120 \times 60 \times 512$ because the ratio of most objects in MOT dataset is approximately equal to it. And finally, the features strengthened by spatial-attention and channel-attention are reshaped to 1024 dimensions to describe the tracking object.

During training, the frames from the same video are fed into network orderly, but the saved features in tracking memory are randomly shuffled before calculating temporal attention scores every iteration. Limited by the GPU memory, we set the maximum length of each trajectory K to 3. We apply stochastic gradient descent(SGD) with the momentum of 0.9 to train the network and the weight decay is set to $5e-5$. The model is trained for 30 epochs with learning rate $1e-5$ in top five epochs, $1e-4$ for the following twenty epochs, and again the former with the last five epochs. For the tracking loss, the threshold σ to punish negative samples is set to 0.5. Online hard example mining(OHEM) and focal loss are used to address the imbalance of positive and negative problems, and data augmentation is used to get better generalization ability.

During inference, the K is set to 5 and σ is the same as training stage. Additionally, a motion model described in IV-E is used to select candidates previously. After the similarity scores between objects and trajectories on the current frame are calculated, the association is then solved by applying the Hungarian algorithm for those greater than σ, and the rest are treated as a new trajectory. Following [14], we applied Non-maximum Suppression(NMS) to the detection results before tracking.

D. performance on the MOT benchmarks

We evaluate our proposed method on the test set of MOT17 benchmark and compare it with several state-of-the-art tracking methods. All the compared state-of-the-art methods use the same public detections provided by the benchmark as us for a fair comparison, and the quantitative results are shown in Table I.

As shown in Table I, our proposed method achieves a comparable performance against the state-of-the-art methods. In terms of MOTA, which is the most important metric for MOT, we achieve the best performance among all the online methods and it is also on par with the state-of-the-art offline methods. Compared with FPSN, which have the similar network architecture with us but without multi-attention structure, we improve 3.6% in MOTA, 1.2% in MT, having fewer samples in FP and FN, and more importantly, we halve the IDSw with the help of attention mechanism. On the other hand, compared with DMAN, which combines a different attention mechanism with a state-of-the-art Single Object Tracker together, we still obtain little advantage on

TABLE II
CONTRIBUTIONS OF EACH COMPONENT

Experiments	motion model	RFModule	Multi-Attention	MOTA
Exp1				46.1%
Exp2	✓			47.9%
Exp3	✓	✓		47.6%
Exp4	✓		✓	48.8%
Exp5	✓	✓	✓	49.4%

MOTA. And we achieve a great balance between various metrics using a simple network architecture without the help of the state-of-the-art Single Object Tracker, which results in some disadvantages in IDSw honestly.

E. ablation studies

To demonstrate the contribution of each module in our proposed method, we set up several experiments for components of different aspects of our approach, the detail of each experiment are described as follows:

- Exp1: directly using ResNet50 with ROI-Align to calculate the similarity score without proposed RFModule, and Multi-Attention mechanism, which is considering as the baseline algorithm.
- Exp2: adding the motion model based Exp1.
- Exp3: adding the proposed RFModule based Exp2.
- Exp4: adding the proposed Multi-Attention mechanism based Exp2.
- Exp5: adding both the RFModule and Multi-Attention based Exp2, which include all of our proposed components.

Table II shows the performance of all the experiments on the training set of MOT17 dataset. As we can see, all proposed components make contributions to the final performance in terms of MOTA. Comparing Exp2 and Exp3, when we add RFModule separately based on our baseline algorithm, the MOTA gets slightly dropped. however, the RFModule can achieve a significant improvement combined with the proposed Multi-Attention mechanism, Comparing Exp4 and Exp5. Finally, the proposed Multi-Attention mechanism which improves the MOTA score 2.7% separately makes the most important contributions obviously.

Fig.4 shows the visualization of the proposed Multi-Attention. In Fig.4(a), we list four groups examples, and each group consists of three images. The first image cropped from video frame shows its original appearance, and the last two show the extracted features before and after spatial and channel attention network. Compared with each other, our network enhances the discrimination of features by focusing on visible local regions. In Fig.4(b), we present the weight score calculated from temporal-attention in some trajectories. As we can see, the network assigns a lower score to the occluded object in order to reduce their suppression. These examples demonstrate the effectiveness of the proposed Multi-Attention network.

VI. Conclusion

In this paper, we have proposed an online MOT approach via multiple attention mechanism(Multi-Attention) to handle the frequent interactions between targets. For a single object, the network can adaptively focus on learning features in visible local regions to get a more recognizable object descriptor. And for multiple objects across frames, the network can also pay different attention to different objects to suppress the effect of occlusion. By using the Multi-Attention mechanism, we try to handle the occlusion problem in MOT from two dimensions of space and time. And the experimental results on MOTChallenging benchmarks demonstrate the effectiveness of the proposed online MOT algorithm.

Acknowledgment

This work was supported in part by the Natural Science Foundation of China under Grant U1536203 and 61972169, in part by the National key research and development program of China (2016QY01W0200), in part by the Major Scientific and Technological Project of Hubei Province (2018AAA068 and 2019AAA051).

References

[1] Leal-Taixé, Laura, et al. ”Motchallenge 2015: Towards a benchmark for multi-target tracking.” arXiv preprint arXiv:1504.01942 (2015).

[2] Milan, Anton, et al. ”MOT16: A benchmark for multi-object tracking.” arXiv preprint arXiv:1603.00831 (2016).

[3] Berclaz, Jerome, et al. ”Multiple object tracking using k-shortest paths optimization.” IEEE transactions on pattern analysis and machine intelligence 33.9 (2011): 1806-1819.

[4] Chari, Visesh, et al. ”On pairwise costs for network flow multi-object tracking.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[5] Chen, Jiahui, et al. ”Enhancing Detection Model for Multiple Hypothesis Tracking.” 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) IEEE, 2017.

[6] Tang, Siyu, et al. ”Subgraph decomposition for multi-target tracking.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[7] Tang, Siyu, et al. ”Multi-person tracking by multicut and deep matching.” European Conference on Computer Vision. Springer, Cham, 2016.

[8] Zhou, Xiaoqin, et al. ”Multi-channel features spatio-temporal context learning for visual tracking.” IEEE Access 5 (2017): 12856-12864.

[9] Bochinski, Erik , V. Eiselein , and T. Sikora . ”High-Speed tracking-by-detection without using image information.” 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS) IEEE, 2017.

[10] Leal-Taixé, Laura, Cristian Canton-Ferrer, and Konrad Schindler. ”Learning by tracking: Siamese CNN for robust target association.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2016.

[11] Chanho Kim, et al. ”Multiple Hypothesis Tracking Revisited.” 2015 IEEE International Conference on Computer Vision (ICCV) IEEE, 2015.

[12] Lee, Sangyun, and Euntai Kim. ”Multiple object tracking via feature pyramid Siamese networks.” IEEE Access 7 (2018): 8181-8194.

[13] C. Kim, F. Li, and J. M. Rehg. ”Multi-object tracking with neural gating using bilinear lstm.” In Proceedings of the European Conference on Computer Vision (ECCV), pages 200–215, 2018.

[14] Xu, Yihong, et al. ”DeepMOT: A Differentiable Framework for Training Multiple Object Trackers.” arXiv preprint arXiv:1906.06618 (2019).

[15] Xu, Jiarui, et al. ”Spatial-Temporal Relation Networks for Multi-Object Tracking.” arXiv preprint arXiv:1904.11489 (2019).

[16] Chu, Peng, and Haibin Ling. ”FAMNet: Joint Learning of Feature, Affinity and Multi-dimensional Assignment for Online Multiple Object Tracking.” arXiv preprint arXiv:1904.04989 (2019).

[17] Lee, Seong-Ho, Myung-Yun Kim, and Seung-Hwan Bae. ”Learning discriminative appearance models for online multi-object tracking with appearance discriminability measures.” IEEE Access 6 (2018): 67316-67328.

[18] Xiang, Yu, Alexandre Alahi, and Silvio Savarese. ”Learning to track: Online multi-object tracking by decision making.” Proceedings of the IEEE international conference on computer vision. 2015.

[19] Zhu, Ji, et al. ”Online multi-object tracking with dual matching attention networks.” Proceedings of the European Conference on Computer Vision (ECCV). 2018.

[20] Breitenstein, Michael D., et al. ”Online multiperson tracking-by-detection from a single, uncalibrated camera.” IEEE transactions on pattern analysis and machine intelligence 33.9 (2010): 1820-1833.

[21] Yan, Xu, et al. ”To track or to detect? an ensemble framework for optimal selection.” European Conference on Computer Vision. Springer, Berlin, Heidelberg, 2012.

[22] Chu, Peng, et al. ”Online multi-object tracking with instance-aware tracker and dynamic model refreshment.” 2019 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, 2019.

[23] Lin, Tsung-Yi, et al. ”Focal loss for dense object detection.” Proceedings of the IEEE international conference on computer vision. 2017.

[24] Choi, Wongun. ”Near-online multi-target tracking with aggregated local flow descriptor.” Proceedings of the IEEE international conference on computer vision. 2015.

[25] Wang, Fei, et al. ”Residual attention network for image classification.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017.

[26] Hu, Jie, Li Shen, and Gang Sun. ”Squeeze-and-excitation networks.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[27] Woo, Sanghyun, et al. ”Cbam: Convolutional block attention module.” Proceedings of the European Conference on Computer Vision (ECCV). 2018.

[28] He, Kaiming, et al. ”Deep residual learning for image recognition.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[29] He, Kaiming, et al. ”Mask r-cnn.” Proceedings of the IEEE international conference on computer vision. 2017.

[30] Szegedy, Christian, et al. ”Inception-v4, inception-resnet and the impact of residual connections on learning.” Thirty-First AAAI Conference on Artificial Intelligence. 2017.

[31] Felzenszwalb, Pedro F., et al. ”Object detection with discriminatively trained part-based models.” IEEE transactions on pattern analysis and machine intelligence 32.9 (2009): 1627-1645.

[32] Ren, Shaoqing, et al. ”Faster r-cnn: Towards real-time object detection with region proposal networks.” Advances in neural information processing systems. 2015.

[33] Yang, Fan, Wongun Choi, and Yuanqing Lin. ”Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.