IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 29, 2018, accepted December 14, 2018, date of publication December 21, 2018,
date of current version January 11, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2889187

Recurrent Metric Networks and Batch Multiple Hypothesis for Multi-Object Tracking

LONGTAO CHEN[1], XIAOJIANG PENG[2], AND MINGWU REN[1]
[1]School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China
[2]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China

Corresponding author: Mingwu Ren (mingwuren@163.com)

This work was supported by the National Natural Science Foundation of China under Grant 61727802.

ABSTRACT Multi-object tracking aims to recover the object trajectories, given multiple detections in video frames. Object feature extraction and similarity metric are the two keys to reliably associate trajectories. In this paper, we propose the recurrent metric network (RMNet), a convolutional neural network-recurrent neural network-based similarity metric framework for the multi-object tracking. Given a reference object, the RMNet takes as input random positive and negative detections and outputs similarity scores over time. The RMNet handles the long-term temporal object variations and false object detections by its long–short memory units. With the scores from RMNet, we introduce a batch multiple hypothesis (BMH) strategy, a simple yet efficient data association method for the batch multi-object tracking. BMH generates a hypothesis tree for each object with a dual-threshold hypothesis generation approach and, then, selects the best branch (or hypothesis) for each object as the batch tracking result. Specially, we model the hypothesis selection as a 0-1 programming problem and introduce a reward function to re-find the objects in case of missing detection. We evaluate our RMNet and BMH strategy on several popular datasets: 2DMOT2015, PETS2009, TUD, and KITTI. We achieve a performance comparable or superior to those of the state-of-the-art methods.

INDEX TERMS Multi-object tracking, pedestrians tracking, recurrent metric networks.

I. INTRODUCTION

Multiple Object Tracking (MOT) in videos is an important computer vision task and has attracted increasing attention due to its wider range of applications such as visual surveillance, activity analysis [1]–[3], autonomous driving [4] and robot navigation [5], [6]. The challenges of multi-object tracking include occlusions, large intra-object variations, multi-object interactions, etc. Objects to track can be pedestrians on the street, vehicles on the road, sport players on the court and so on. We mainly focus on pedestrian tracking in this paper since pedestrians are ideal non-rigid examples to study the MOT problem.

The most popular pipeline of MOT is the tracking-by-detection or detection-based tracking (DBT), where objects are first detected/identified by an object detector in each frame and then associated with object trajectories across video frames. Generally, two major issues should be considered for a MOT approach. One is how to measure the similarity between objects and detections in frames, the other one is how to recover the identification information based on the similarity measurement. The first issue involves feature extraction and similarity metric learning. The second one involves the inference problem or data association. For the first issue, traditional methods [7]–[9] mainly use two categories of features, namely motion features and appearance features, such as HOG [10], HOF [11], SIFT [12], LBP [13], color histogram. With the features, traditional methods employ approaches like SVM [14], or logistic regression for metric learning.

Recently, deep learning has achieved great success in many vision tasks such as image classification [15], object detection [16], image segmentation [17], and parsing [18]. Some attempts of convolutional neural networks (CNN) have been made into single object tracking [19] and multi-object tracking [20]–[22]. These methods either apply CNN as a powerful feature extractor [22] or further use CNN as a siamese network for similarity metric learning [20], [21]. These methods mainly consider the previous nearest frames to update the extractor or the siamese network, which is not robust for error or missing detections. Intuitively, the

long-term temporal information (i.e., a long sequence) is helpful for both similarity computation and data association. Milan *et al.* [23] present an end-to-end Recurrent Neural Network (RNN) for the data association of MOT, which uses RNN to encode temporal information.

In this paper, to take full advantage of long-term information, we propose RMNet for similarity metric learning, and a batch multiple hypothesis (BMH) strategy for data association. The RMNet is based on a convolutional neural network with the head of some long-short term memory units (CNN-LSTM). For our RMNet, CNN is used for feature extraction, and LSTM is used for similarity metric learning with long-term temporal detections. Given a reference object and a batch of frames, the RMNet takes as input random temporal negative and positive detections and outputs similarity scores (from 0 to 1) over time. Unlike siamese networks for metric learning, the RMNet handles the long-term temporal object variations and false object detections by its long-short memories naturally, which is important for multi-object tracking.

The BMH is partly inspired by the tree structure of track-oriented multi-hypothesis tracking (TOMHT) [24] for radar objects. Instead of building trees with multiple complicated conditions in TOMHT, our BMH utilizes a dual-threshold hypothesis generation approach with the high-quality similarity scores of RMNet. After building hypothesis trees for each object, our BMH then selects the best branch (or hypothesis) for each object as the tracking result. Specially, given all the object hypothesis trees, we model hypothesis selection as a 0-1 programming problem to select the best hypothesis for every object. To deal with missing detection, we also introduce a reward function to re-find objects in our BMH.

Our contributions can be summarized as follows. First, we propose the RMNet for both feature extraction and similarity measurement which is robust to object variation and false detection. Second, based on the high-quality similarity scores, we introduce a simple yet efficient batch multiple hypothesis strategy for data association of multi-object tracking. Finally, we conduct extensive experiments on several popular datasets, namely 2DMOT2015 [25], PETS2009 [26], TUD [27], and KITTI [28], and achieve performance comparable or superior to these of the state-of-the-art methods.

The remainder of this paper is organized as follows. In Sec. II, we briefly review related work on multi-object tracking. We offer an overview of our multi-object tracking system in Sec. III. We introduce the recurrent metric network and batch multiple hypothesis strategy in Sec. IV and Sec. V. We present experimental results in Sec. VI and conclude our paper in Sec. VII.

II. RELATED WORK

A. MULTI-OBJECT TRACKING

Multi-object tracking can be roughly divided into two categories, namely online multi-object tracking [29]–[31], [31]–[39] and batch multi-object tracking [7], [40]–[46]. The difference is whether or not observations from future frames are utilized when handling the current frame. Online, also called causal, tracking methods only rely on the past information available up to the current frame. Generally, online methods solve the data association problem either probabilistically [35]–[37] or determinatively (e.g., Hungarian algorithm [47] in [31], [38] or greedy association [39]). Batch tracking approaches employ observations both in the past and in the future. Widely-used schemes include global optimization [45], delay decision [48], etc. For global optimization, the most popular algorithm is graph based path searching like K-shortest path(KSP) [42], push-relabel based network flow [46], successive shortest path [45]. He *et al.* [49] developed a connected component model to solve the multi-dimensional assignment problem for multi-object tracking. Another important idea of global optimization is tracklet based association [41], [50], which formulates global optimization as a maximum a posterior problem (MAP). Unlike global optimization, delay decision uses adjacent future information instead of all information. The widely-used delay decision method is the multi-hypothesis tracking (MHT) proposed by Reid *et al.* [48] for radar target tracking.

B. MULTI-OBJECT TRACKING WITH DEEP NEURAL NETWORKS

With the great success of deep convolutional neural networks (CNN) in the computer vision community, some multi-object tracking methods have applied CNN based features [22], [51] or Siamese CNN based similarity metric [20], [21]. In addition, some works utilize recurrent neural networks (RNN) for multi-object tracking. Anton *et al.* design an end-to-end RNN (LSTM) net as the tracker, which utilizes long-short term memory units (LSTM) with a cascade structure [23]. Kuan *et al.* propose the Recurrent Autoregressive Network (RAN), a temporal generative modeling framework to represent the appearance and motion dynamics of multiple objects over time [52]. Due to the limited number of ground-truth trajectories of current multi-object tracking datasets, the performance of RNN can be degraded by inadequate training. Our recurrent metric network differs from these deep learning based methods from that 1) we use the CNN-LSTM architecture for similarity metric learning and bounding box regression, 2) we randomly select temporal positive and negative samples to enlarge the training set of the RMNet instead of only same object samples, 3) the output similarity scores are further fed into our BMH-based tracker.

III. OVERVIEW OF OUR MOT SYSTEM

Figure 1 provides an overview of our multi-object tracking system. It mainly consists of two blocks, namely the batch multi-hypothesis (BMH) tracker and RMNet.

Suppose we have object detections in each frame of a batch as shown in Figure 1(a) (we add labels for better visualization), the BMH tracker utilizes all the possible links between frames and builds a hypothesis (i.e., trajectory) tree

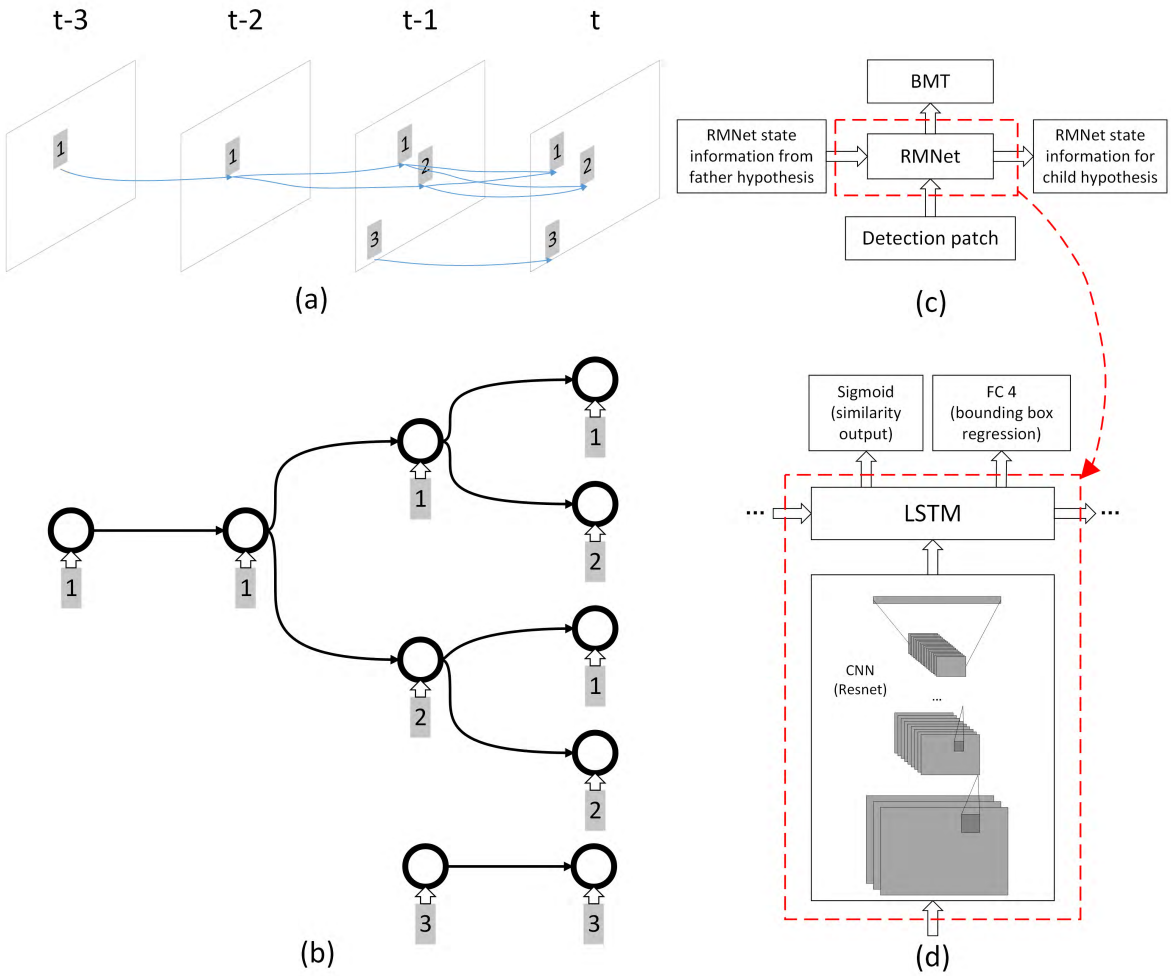


FIGURE 1. Overview of our MOT system. (a) An instance of video sequence. Gray patches are the detections. Blue lines indicate the possible trajectories. (b) The hypothesis tree formed according to the detections in (a). (c) The processing pipeline for a RMNet instance. (d) The architecture of RMNet. RMNet processes the detection patch and provides the similarity and box regression information to BMH.

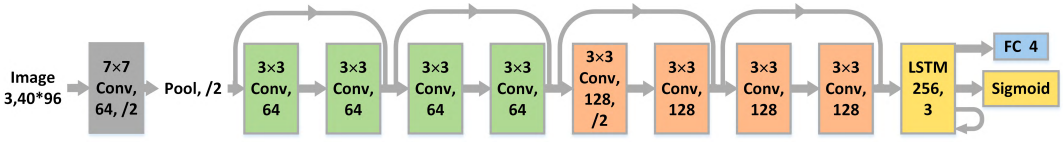


FIGURE 2. The detailed architecture of RMNet.

(see Figure 1(b)) for the batch, and then computes hypothesis scores, and finally generates the target trajectory. The details of BMH are presented in Section V.

One key of a MOT system is the similarity metric between detections and target object. To this end, we integrate the RMNet into BMH as the similarity measurement. At a certain time t, the RMNet takes as input one of the current detections and the state from the parent node of a hypothesis (see Figure 1(c)). The RMNet adjusts the current detections by a bounding box regressor and computes similarity scores for each of them (see Figure 1(d)). Finally, the BMH block adds these scores to the hypothesis tree as child nodes.

IV. THE RECURRENT METRIC NETWORK

We design RMNet for similarity measurement. The overview of the RMNet is depicted in Figure 1(d), which consists of a ResNet [15], a LSTM block, and two output branches. The detailed architecture of RMNet is shown in Figure 2. At a certain time t, the RMNet takes as input an object detection with the previous state of LSTM, and then feeds the detection into a CNN model, i.e., ResNet18, for feature extraction, and then further feeds the features into LSTM units, and finally outputs a sigmoid score and a 4-D vector. Specially, the sigmoid score indicates the similarity between the input detection and the reference object, and the vector refers to

the offset between the detected bounding box (bbox) and the ground-truth bbox.

A. TRAINING WITH VARIOUS SEQUENCES

The head of our RMNet (i.e., LSTM) is a type of recurrent neural network, which is able to deal with long-term sequences. Long-term information can be helpful for tracking. For training our RMNet, a straightforward strategy is to use same object sequences like [52] and [53]. In [52] or [53], a network is proposed for similarity measurement, which compares a known trajectory sequence with a new detection. The known trajectory sequence is composed of bboxes from the same ID when training the network. In fact, it's impossible to guarantee all detections in a known trajectory sequence are from the true target cause mistake is unavoidable during tracking. In practical tracking, the known trajectory sequence could be mixed with improper detections.

Different from the above-mentioned methods, our RMNet aims to jointly learn features and similarity metric. Training with only same object sequences is inappropriate in our observation since we need to separate different objects. To this end, we train our RMNet with various sequences which not only includes the same object sequences but also those sequences with random positive and negative detections (see Figure 3(a)). We state the two main advantages of this training strategy as follows.

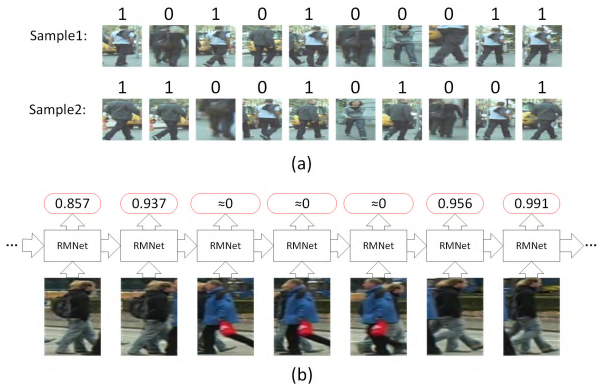


FIGURE 3. (a) Two training sequences. (b). A sequence with mismatches and its detection-level similarities from RMNet.

Learning to resist mismatches. Generally, using improper data (inaccurate or wrong detections) to update the similarity metric model corrupts the whole tracking system. We refer the improper data as 'mismatches' in this paper. Traditional methods deal with mismatches mainly by exploiting all kinds of information and adding complex constraints for each object model to avoid it. Compared to the traditional methods, our strategy, i.e., learning to resist mismatches by the model itself, is a simple yet efficient one. And it can be shared by all tracking objects. With various sequences, our RMNet is able to learn discriminative features and metric model without worrying mismatches. Figure 3(b) shows an example with our RMNet. The RMNet outputs very low similarities for mismatches (see the middle three detections) and keeps high ones after those mismatches.

Data augmentation. Our training strategy, i.e., training with various sequences, can be seen as a kind of data augmentation. For tracking task, the limited number of training data is an unavoidable problem. Common data augmentation methods like flipping, cropping, and color jittering can be used to generate similar trajectories. Milan *et al.* [23] propose a physically-based trajectory generation method to enlarge training data, where they only use location information for training. By randomly selecting detections as trajectories, we are able to enlarge the training sequences exponentially. In addition, we also employ common data augmentation approaches.

B. BOUNDING BOX REGRESSION

To alleviate the impact of inaccurate detections, we use a bounding box regressor to refine the detections. Following [54], We use smooth L1 loss for regression as follows,

$$L_b = \sum_{i\in\{x,y,w,h\}} smooth_{L_1}(t_i - t_i^*), \tag{1}$$

where

$$smooth_{L_1}(x) = \begin{cases} 0.5x^2 & if\ |x| < 1 \\ |x| - 0.5 & otherwise, \end{cases} \tag{2}$$

and

$$\begin{aligned} t_x &= (x - x_a)/w_a, \quad t_y = (y - y_a)/h_a, \\ t_w &= \log(w/w_a), \quad t_h = \log(h/h_a), \\ t_x^* &= (x^* - x_a)/w_a, \quad t_y^* = (y^* - y_a)/h_a, \\ t_w^* &= \log(w^*/w_a), \quad t_h^* = \log(h^*/h_a) \end{aligned} \tag{3}$$

where x, y denote the center coordinates of a box, and w, h denote the width and height. Variables x, x_a, and x^* are for the regressed bbox, detection bbox, and ground-truth bbox, respectively, likewise for y, w, h. The latter two kinds of boxes are provided by the multi-object tracking dataset itself. This step adjusts the provided detections to the corresponding ground-truth bbox.

C. LOSS FUNCTION OF RMNET

For training RMNet, we use two kinds of loss functions, namely the binary cross entropy loss and the smooth L1 loss (see Eq. (1)). The overall loss is as follows,

$$L = L_s + \omega_b L_b, \tag{4}$$

where ω_b is the trade-off weight for bounding box regression loss which is fixed as 10 empirically, and L_s is the binary cross entropy loss which is formulated as follows,

$$L_s = \frac{\sum_{i\in\{1,2,3...N\}} -(y_i \log x_i + (1 - y_i)\log(1 - x_i))}{N}, \tag{5}$$

where N is the length of the sequence and y_i, x_i indicate target and sigmoid output, respectively. The sigmoid output refers

to the similarity score between an input detection patch and a target object. Figure 3(a) shows two samples with target binary labels where 1 indicates the same object and 0 the different one.

V. BATCH MULTI-HYPOTHESIS STRATEGY FOR MULTI-OBJECT TRACKING

BMH is a tree structure based tracker which outputs batch decision. In a certain batch, it associates the detections to a hypothesis tree, and aims to select the best hypothesis (one of the branches) yet prune the others simultaneously. In the next batch, only one hypothesis for each reference object is preserved and used as the root for further tree building. This strategy includes two key stages, i.e., hypothesis generation and selection. For hypothesis generation, we propose a dual-threshold hypothesis generation approach to perform the association and decide how the tree is built, which mainly utilizes the information of RMNet. As for selection, we model the hypothesis selection as a 0-1 programming problem, where mutual exclusion is embedded as a constraint.

For the convenience of expression, we introduce the notations used in this section first. The set of detections received at frame t is denoted by $D^t = \{d_i^t\}_{i=0}^{M_t}$, where M_t is the number of detections at frame t. A hypothesis H_j^t at frame t is defined as a sequence of detections: $H_j^t = (d_{j_1}^1, d_{j_2}^2, ..., d_{j_t}^t, ..., d_{j_k}^k)$, $d_{j_t}^t \in D^t$, note $d_{j_t}^t$ can be a dummy detection in the case a miss detection occurs at frame t. H^t is the set of hypothesis at frame t: $H^t = \{H_i^t\}_{i=0}^{L_t}$, where L_t is the number of hypotheses at frame t. In the following, we present the definition of hypothesis score, our hypothesis generation approach, our re-find reward function, and the mechanism of hypothesis selection.

A. HYPOTHESIS SCORE

To indicate the likelihood of a hypothesis, each hypothesis is assigned with a hypothesis score, which is updated at each time step. We define the score $S(H_j)$ of hypothesis H_j as the weighted sum of two terms, i.e., motion score S_{mot} and appearance score S_{app}. Following [48], we utilize S_{mot} as the motion term to indicate the association potential with respect to object motion. This motion equation is employed in TOMHT for tracking point object and proved to be effective in capturing motion information. In addition, we introduce an appearance term S_{app} based on the outputs of RMNets to measure the appearance similarity according to the associations of hypothesis path.

The score of hypothesis j at frame t is defined as Eq. 6:

$$S(H_j^t) = S_{mot}(H_j^t) + S_{app}(H_j^t), \tag{6}$$

where $S_{mot}(H_j^t)$ is defined as follows,

$$S_{mot}^t = \sum \Delta S_{mot}^i(H_j^t). \tag{7}$$

We experimentally find that the final performance is not sensitive to the ratio of S_{mot} and S_{app}. If hypothesis H_j^{t-1} is associated with detection d_i^t at frame t the increment of hypothesis score is given by,

$$\Delta S_{mot}^i(H_j^t) = \begin{cases} \log\left(\dfrac{p(d_i^t|H_j^{t-1})P_D}{\lambda_{fa}+\lambda_{nt}}\right) & i \neq 0, \\ \log(1-P_D) & i = 0. \end{cases} \tag{8}$$

where the notations refer to the follows,
$p(\cdot)$: the probability density function (PDF) of detection d_i^t conditioned on the one-step prediction of hypothesis H_j^{t-1};
P_D : detection probability;
λ_{fa} : the expected number of false alarms per unit volume of the detection space per frame (spatial density of clutter);
λ_{nt} : spatial density of new targets.
the initial hypothesis score $\Delta S_{mot}^i(H_j^t)$ is given as $\log(\frac{\lambda_{nt}}{\lambda_{fa}})$.

Then, we define $S_{app}(H_j^t)$ as follows,

$$S_{app}^t = \sum \Delta S_{app}^i(H_j^t) = \sum \log\left(\frac{P(D \in H)}{P(D \notin H)}\right), \tag{9}$$

where $P(D \in H)$ is given according to the sigmoid output of RMNet, and $P(D \notin H)$ is set as $1 - P(D \in H)$.

B. SIMILARITY BASED HYPOTHESIS GENERATION

We design a dual-threshold hypothesis generation approach for hypothesis gating and forming. Specifically, two pre-defined thresholds are used for gating, namely threshold $T_{confirm}$ for detection confirming and T_{miss} for missing detection checking. This approach reasons between object and detection when managing hypotheses. For instance, if all associations between a hypothesis and the possible detections are weak, we reason out that this object may be missed by the detector.

Firstly, we provide some notations used in Alg. 1. In Ln. 2, S_{RMNet}^{max} is the maximum similarity score associated to d_i^t. In Ln. 5, $MahalDist(Predict(H_j^{t-1}), d_i^t)$ returns mahalanobis distance between predicted location and detection location. We denote $T_{MaxMaha}$ as the threshold of mahalanobis distance.

Alg. 1 comprises of three parts in three for-loop structures respectively:

1) ASSOCIATION BETWEEN DETECTION AND HYPOTHESIS (LN.1-21)

Only associations with $MahalDist(Predict(H_j^{t-1}), d_i^t)$ lower than $T_{MaxMaha}$ are considered as potential ones (Ln. 5). Given state parameters h, c of LSTM from H_j^{t-1} and image patch $I_{d_i^t}$ of d_i^t, RMNet makes the update and outputs the response $S_{RMNet}(H_j^t)$ (Ln. 6). For all possible associations during tracking, those hypotheses with similarity scores below T_{miss} are discarded directly (Ln. 8). Otherwise, the tracker will accept it as the candidate hypothesis (Ln. 10). In addition, for all hypotheses associated with the same detection, we check whether or not the highest similarity score (i.e., S_{RMNet}^{max} in Ln. 11) exceeds $T_{confirm}$ (Ln. 18). If not, we assume that this detection does not get appropriate reasoning, a new object deriving from this detection would be created (Ln. 19).

Algorithm 1 dual-Threshold Hypothesis Generation

Require: $D^t, H^{t-1}, T_{MaxMaha}, T_{miss}, T_{confirm}$
Ensure: H^t
1: **for** each $d_i^t \in D^t$ **do**
2: $S_{RMNet}^{max} \leftarrow -\inf$
3: $N^i \leftarrow 0$
4: **for** each $H_j^{t-1} \in H^{t-1}$ **do**
5: **if** $MahalDist(Predict(H_j^{t-1}), d_i^t) < T_{MaxMaha}$ **then**
6: $S_{RMNet}(H_j^t) \leftarrow RMNet(h_{H_j^{t-1}}, c_{H_j^{t-1}}, I_{d_i^t})$
7: **if** $S_{RMNet}(H_j^t) < T_{miss}$ **then**
8: discard and continue
9: **end if**
10: create new leaf node H_l^t for H_j^{t-1}, update $S(H_l^t)$ using Eq. 6.
11: $S_{RMNet}^{max} \leftarrow \max(S_{RMNet}(H_j^t), S_{RMNet}^{max})$
12: $N^i \leftarrow N^i + 1$
13: **if** $S_{RMNet}(H_j^t) > T_{confirm}$ and H_j^{t-1} doesn't associate to any detection **then**
14: use Eq. 10 for re-find reward
15: **end if**
16: **end if**
17: **end for**
18: **if** $N^i = 0$ or $S_{RMNet}^{max} < T_{confirm}$ **then**
19: create new object, using d_i^t as start
20: **end if**
21: **end for**
22: **for** each $H_i^t \in H^t$ **do**
23: $H_{best\_child} \leftarrow \underset{H_x \in \{H_x | father(H_x^t) = father(H_i^t)\}}{\arg\max} (S_{RMNet}(H_x^t))$
24: **if** $S_{RMNet}(H_{best\_child}) > T_{confirm}$ and $H_{best\_child} \neq H_i^t$ **then**
25: delete H_i^t
26: **end if**
27: **end for**
28: **for** each $H_j^{t-1} \in H^{t-1}$ **do**
29: **if** H_j^{t-1} has no child and $S_{app}(H_j^{t-1}) > \log \frac{T_{confirm}}{T_{miss}}$ **then**
30: create leaf node H_l^t(without detection) for H_j^{t-1}, update $S(H_l^t)$ using Eq. 6 with $S_{RMNet}(H_l^t) = T_{miss}$.
31: **end if**
32: **end for**

2) HANDLING CONFIRMED ASSOCIATION (LN.22-27)

In Ln. 23, $father(H_x^t)$ returns the hypothesis associated with H_x^t at frame $t-1$.For hypotheses from same father node, we search for the best child (i.e., hypothesis with highest similarity score) and compare its similarity score with $T_{confirm}$ (Ln. 24). If the score is greater than $T_{confirm}$, we assume that current object gets reasonable propagation in this frame. Then, only this best hypothesis would be retained while all others get removed to avoid future mis-alignments (Ln. 25).

3) HANDLING MISSING DETECTION (LN.28-32)

Finally, for those hypotheses fail to find associated detection (or called dummy hypothesis, which is usually caused by missing detection), we preserve them for a short period before deleting. We use $\log \frac{T_{confirm}}{T_{miss}}$ as the threshold for $S_{app}(H_j^{t-1})$ to decide whether or not to delete a dummy hypothesis. For a dummy hypothesis that has no current associated detection, we simply skip the update of RMNet to avoid unnecessary model drift. Then we set the $S_{RMNet}(H_j^{t-1})$ as T_{miss} to update S_{app} and S_{app} will decrease.

C. RE-FIND REWARD

Another problem caused by the uncertainty of detection is missing detection. We try to fill the gap of missing frames by an extra reward. When a hypothesis regains reliable association ($S_{RMNet}(H_j^t) > T_{confirm}$) with a target after missing its previous object, it will be rewarded with a re-find reward (Alg. 1 Ln 14,15). This mechanism can help these hypotheses to survive during selection phase. Because lack of matching for a long time dramatically weakens hypothesis and produces a low score.

$$S_{app}^t = S_{app}^t + N_{miss} \log\left(\frac{T_{confirm}}{T_{miss}}\right) \tag{10}$$

N_{miss} is the number of missed frames where no matching detection is found.

D. HYPOTHESIS SELECTION AS 0-1 PROGRAMMING

We perform hypothesis selection in a batch way. In other words, the best hypothesis from a tree is selected and retained at the frame indicated by the blue dashed line with the equal interval as Figure 4 shows. Then, the selected hypothesis is regarded as the root for next growth. The final result is the path composed of the selected nodes, which is marked by red in Figure 4.

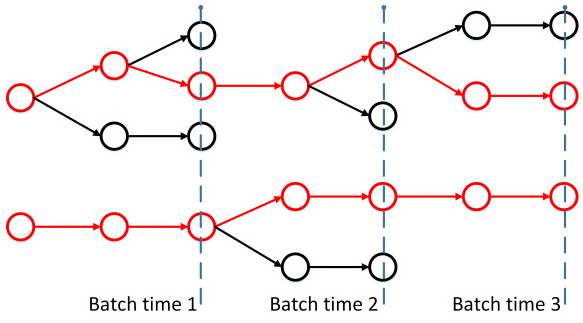


FIGURE 4. The hypothesis is selected according to batch interval. Then, the selected one is retained as the root for next batch. The batch interval in this figure is set to 3 frames for simplicity. Selected path and nodes are marked in red.

Track selection is naturally a binary linear programming problem in the view of treating the selection of a hypothesis as the switch of a binary variable. Besides, the occupancy of detection should be considered carefully, i.e., the mutual exclusion between hypotheses. We employ a popular

assumption, one-to-one association, which insists one detection could only be connected to one track. The formula for hypothesis selection is like following.

$$\min C = \min \sum_n \xi_{H_n} C_{H_n} \tag{11}$$

$$s.t.\ \xi_{H_i} + \xi_{H_j} <= 1, \quad H_i \cap H_j \neq \emptyset \quad and\ i \neq j \tag{12}$$

$$\xi_{H_n} \in \{0, 1\} \tag{13}$$

C_{H_n} is defined as the cost of hypothesis H_n. ξ_{H_i} is a binary representation of H_i. Constraint 12 indicates the relation of mutual exclusion, where if two hypotheses contain the same detection then up to one of them will be selected. We use the following criteria as the cost of a hypothesis.

$$C_{H_n} = -S(H_n^t) \tag{14}$$

where $S(H_n^t)$ is the score of leaf hypothesis H_n at frame t.

Then, after we build the problem of 0-1 programming for each batch, we use lpsolve to solve it. Owning to the rational assumptions and constraints, solving progress is not complex.

E. SUMMARY OF BMH

We briefly summarize BMH strategy in Alg. 2, where N is the maximum frame number in a video. For each time step, we input current detections (D^t) and hypothesis set from last frame (H^{t-1}) into dual-threshold hypothesis generation to build hypothesis tree. At the end of a batch, we then select the best hypothesis (or the trajectory) for every tree via 0-1 programming. Finally, at the end of the video, we refine all trajectories by exerting Kalman smoothing and remove those short trajectories with the number of detected frames less than a threshold.

Algorithm 2 Batch Multi-Hypothesis Strategy

Require: $(D^1, D^2, D^3, ..., D^N), T_{MaxMaha}, T_{miss}, T_{confirm}$
Ensure: H^N
1: $H^0 \leftarrow \emptyset$
2: **for** each $t \in 1, 2, 3, ..., N$ **do**
3: $H^t \leftarrow$ dual-threshold hypothesis generation $(D^t, H^{t-1}, T_{MaxMaha}, T_{miss}, T_{confirm})$
4: **if** t is at batch timing **then**
5: use Eq.11 for hypothesis selection.
6: **end if**
7: **end for**
8: delete hypothesis with too less detections.
9: refine trajectories by exerting Kalman smoothing.

VI. EXPERIMENTS

In this section, we first present the used datasets and implementation details, then show the performance evaluation in common measurements and speed.

A. DATASETS

We perform experiments on the 2DMOT2015 [25], PETS-2009 [26], TUD [27], and KITTI [28] datasets and compare our methods to the state-of-the-art algorithms. **2DMOT2015** is a widely-used yet challenging multi-object tracking dataset, and it consists of 22 video clips with multiple views, camera motions and various weather conditions. 11 videos are used for training and the other 11 videos for testing. This dataset provides detections with the Modified ACF [58] object detector. Ground-truth information is only provided for training set. Thus, as a common protocol, we divide the training set of 2DMOT2015 into two subsets for training and testing, where videos of TUD-Stadmitte, PETS09-S2L1, TUD-Campus and ETH-Pedcross2 are used for testing and the others are used for training. The **PETS_2009_S2** dataset contains three long-term videos which are designed for pedestrian tracking with different densities of pedestrians. Occlusion and illumination are the main challenges of PETS_2009_S2. The **TUD** dataset also contains 3 videos, which are captured in busy street with a low camera angle in close range. Scale variance and occlusion are the main problems in TUD. The **KITTI** dataset consists of about 19,000 frames (32 minutes) or 50 videos which are recorded using cameras mounted on top of a moving vehicle. It is divided into two subsets: 21 training videos and 29 testing videos. Each video sequence has a varied number of frames from 78 to 1176 frames.

B. IMPLEMENTATION DETAILS

Our RMNet is implemented with the PyTorch framework[1] and the BMH tracker is implemented in C++. The implementation is on a PC with a 3.3GHz CPU (4 cores) and a 1080ti GPU.

1) ACCELERATION STRATEGY

In practice, conducting RMNet on all possible hypotheses in a batch is very time-consuming. To this end, we add distance constraints to select hypotheses. Specifically, we first set a maximum searching distance as twice of the width for a reference bbox, and then pre-compute the similarities of possible associations within this distance which can be reused in a batch.

2) PARAMETERS SETTING

For BMH, both $T_{confirm}$ and T_{miss} are set according to the confidence scores on validation set. $T_{confirm}$ is the boundary value to ensure a sample to be positive with higher value, and is set to the value which makes 95% to be positive on the validation set. T_{miss} is set to ensure a sample to be negative with less value, and is set to the value which makes 95% to be negative. $T_{MaxMaha}$ is set to 16. As for the parameters in S_{ori}, we set P_D, λ_{nt}, λ_{fa} as 0.9,1e-8 and 1e-6 respectively. All the detections are resized as $3 \times 96 \times 40$ to fit the network input size. Horizontal flipping and color jittering are

[1] http://pytorch.org/

TABLE 1. Performance comparison between our tracking method and other methods for the 2DMOT2015 dataset. downward arrow indicates lower is better, and the upward arrow indicates higher is better.

Method	ID Measures			CLEAR MOT								
	IDF1(↑)	IDP(↑)	IDR(↑)	FAR(↓)	FP(↓)	FN(↓)	IDSW(↓)	FM(↓)	MOTA(↑)	MOTP(↑)	MOTAL(↑)	Hz(↑)
TC_ODAL [39]	-	-	-	2.24	12970	38538	637	1716	15.1	70.5	16.2	1.7
CEM [7]	-	-	-	2.45	14180	**34591**	813	1023	19.3	70.7	20.6	1.1
ELP [58]	26.2	38.7	19.8	1.27	7345	37344	1396	1804	25.0	71.2	27.3	5.7
SegTrack [8]	31.5	47.6	23.5	1.36	7890	39020	697	**737**	22.5	71.7	23.6	0.2
MotiCon [59]	29.4	39.8	23.3	1.80	10404	35844	1018	1061	23.1	70.9	24.7	1.4
JPDA_m [60]	33.8	54.3	24.5	**1.10**	**6373**	40084	**365**	869	23.8	68.2	24.4	32.6
LP_SSVM [9]	34.0	48.8	26.1	1.45	8369	36932	646	849	25.2	71.7	26.3	41.3
RNN_LSTM [23]	17.1	23.0	13.6	2.00	11578	36706	1490	2081	19.0	71.0	21.4	**165.2**
Ours	**38.7**	**57.5**	**29.2**	1.17	6733	36952	477	790	**28.1**	**74.3**	**28.9**	16.9(300)

used for data augmentation. For training LSTM, we randomly sample fragments from a detection sequence as positives. Then, we insert detections from different objects as negatives. Specifically, we choose those different objects from a neighbor area in nearby frames, which have high probability to be mistracked during tracking process. This neighbor distance is set as $t_{diff}\,V_{max\_speed}$, where t_{diff} is the frame difference and V_{max_speed} is set as 30. With the backbone ResNet-18 pre-trained on ImageNet, we train the RMNet by stochastic gradient descent with mini-batch size of 64. The learning rate is initialized as 0.02 for LSTM and 0.0001 for the ResNet part, and decreased by a factor of 5 every 20 epochs. We stop training at 200 epochs. All the mentioned parameter values are used as default in the following experiments.

C. EXPLORATION OF RMNET ON PUBLIC DATASETS

1) METRICS

We use CLEAR-MOT [59] and ID measures [60] as evaluation metrics. CLEAR-MOT includes the multi-object tracking accuracy (MOTA), multi-object tracking precision (MOTP), etc. MOTA is a score which combines false positives, false negatives and identity switches (IDSW) of the output trajectories. MOTP measures the alignment accuracy between the positive trajectories and the ground truth trajectories in terms of the average distance. In addition to these metrics, the average number of false alarms per frame (FAR), the total number of false positives (FP), the total number of false negatives (FN), track fragmentations (FM), and IDSW are also reported. We use the IDP, IDR, IDF1 counts to compute identification precision (IDP), identification recall (IDR), and the corresponding F1 score IDF1.

2) EVALUATION ON 2DMOT2015

Table 1 presents the average performance over all test videos. With the ID measures, our method achieves state-of-the-art results on 2DMOT2015. With the CLEAR-MOT metrics, our method obtains comparable performance to recent approaches. The ID measures are performed by one-to-one matching between tracked objects and ground truths, which includes re-tracked objects. However, the CLEAR-MOT metrics refer re-tracked objects as errors which is relatively unfair for batch multi-object tracking since we have re-tracking strategy. Compared to RNN_LSTM [23] which also utilizes LSTM for tracking, our method achieves significantly better performance in MOTA and IDF, namely 19 vs 28.1 and 17.1 vs 38.7. It is worth noting that the evaluation on 2DMOT2015 is more convincing than the following relatively small datasets.

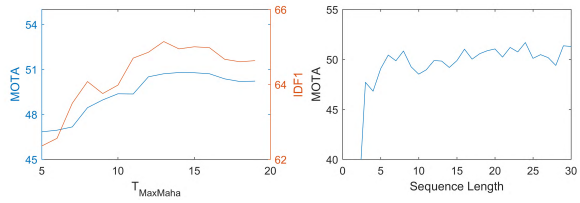


FIGURE 5. Evaluation of $T_{MaxMaha}$ (left) and the input length of RMNet (right).

3) EVALUATION ON PETS_2009, TUD, AND KITTI

For this evaluation, we use the same model and parameters as for 2DMOT2015 to verify our method on these datasets. Table 2, Table 3, and Table 4 show the comparison between our method and several popular algorithms. The PETS_2009 S2L1 sequence is a long and low-pedestrian-density sequence with 795 frames. The PETS_2009 S2L2 sequence is a medium-crowded sequence

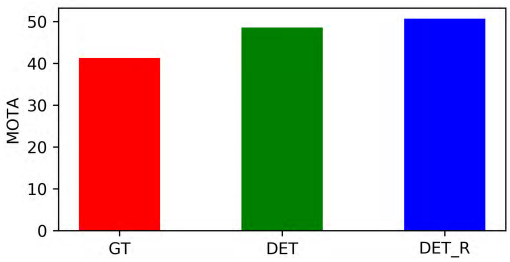


FIGURE 6. Comparison of training with different bounding boxes and w/wo regression. GT: ground-truth boxes. DET: detected boxes without regression. DET_R: detected boxes with regression.

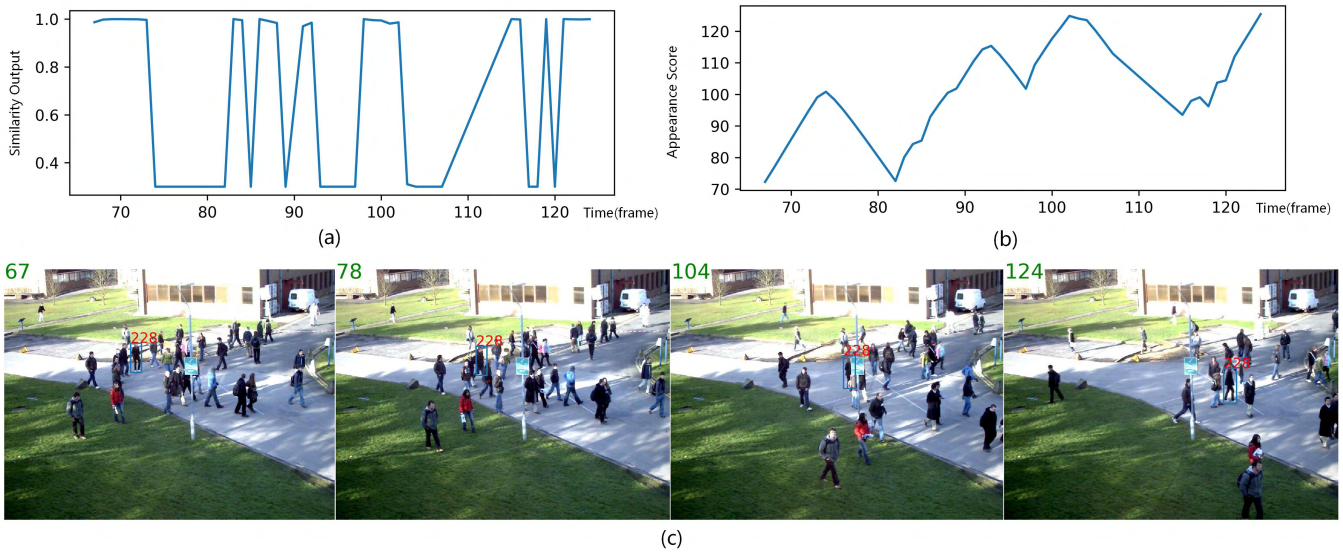


FIGURE 7. An instance of track segment from PETS-S2L2. (a)Curve of similarity output from RMNet; (b)Curve of S_{app}; (c)Four representative moments indicate normal, missing, mismatch and re-tracking respectively.

TABLE 2. Performance comparison on the PETS 2009 dataset.

Sequence	Method	FP(↓)	FN(↓)	IDSW(↓)	MOTA(↑)	MOTP(↑)
PETS2009 S2L1	RMOT [29]	2485	417	41	36.7	65.3
	SCEA [30]	1890	1903	3	18.4	**90.4**
	CMOT [31]	1435	572	37	83.04	69.59
	CEM* [7]	59	302	11	90.6	80.2
	KSP* [43]	126	641	13	80.3	72.0
	H^2T* [45]	62	222	11	82.7	72.9
	CDA_DDALpb [34]	-	-	4	**91.0**	80.5
	Ours	293	213	11	88.4	79.3
PETS2009 S2L2	RMOT [29]	4326	563	190	37.2	67.7
	SCEA [30]	1859	5504	760	21.1	59.9
	CMOT [31]	2422	4860	45	**60.1**	53.9
	SMOT* [42]	-	-	251	34.4	70.0
	DCT* [41]	-	-	-	37.5	56.4
	CEM* [7]	622	2881	150	44.9	59.4
	KSP* [43]	193	6117	22	24.2	60.9
	CDA_DDALpb* [34]	-	-	113	50.8	70.3
	Ours	438	3771	162	54.7	**75.6**

TABLE 3. Performance comparison on the TUD dataset.

Sequence	Method	FP(↓)	FN(↓)	IDSW(↓)	MOTA(↑)	MOTP(↑)
TUD Stadtmitte	RMOT [29]	784	487	6	-	56.8
	SCEA [30]	368	529	6	21.9	56.9
	CMOT [31]	459	495	9	16.7	56.8
	CEM* [7]	92	108	4	71.1	65.5
	KSP* [43]	117	261	5	45.8	56.7
	DCT* [41]	-	-	4	61.8	63.2
	DTLE* [44]	134	457	15	56.2	61.6
	ISR_RDE [64]	167	44	11	**80.8**	**81.7**
	Ours	0	258	2	77.5	80.0
TUD Campus	RMOT [29]	49	4	15	**83.2**	91.9
	SCEA [30]	41	41	0	79.8	**99.9**
	CMOT [31]	21	92	0	72.1	99.1
	SMOT* [42]	5	173	5	54.8	94.9
	ISR_RDE [64]	62	17	17	74.8	94.9
	Ours	3	94	3	72.1	76.7

with 436 frames. Large appearance variations are caused by illumination and occlusion in PETS_2009. Generally, our method achieves performance comparable to the state of the arts in all measurements. CDA_DDALpb [34] is also based on deep appearance learning, which aims to learn a discriminative appearance model from large training datasets and improve appearance discriminability by adapting the pre-trained deep model during online tracking. Our method outperforms CDA_DDALpb on the PETS_2009 S2L2 sequence in both MOTA and MOTP.

The TUD-Stadtmitte and TUD-campus in TUD dataset are real-world video sequences with occlusions and interactions between pedestrians. Again, without optimization, we obtain performance comparable to the state of the arts according to MOTA and IDSW which are the most important two measurements for the MOT task.

In Table 4, we present the results on KITTI pedestrian test dataset [28] with the same model trained on 2DMOT2015. Without optimization on KITTI, our model still outperforms several other methods in MOTA. From the test on the above three small datasets, our model trained on 2DMOT2015 is robust to PETS_2009, TUD, and KITTI.

D. EVALUATION OF HYPER-PARAMETERS

In this section, we first evaluate several hyper-parameters introduced by our method, namely i) the Mahalanobis distance threshold $T_{MaxMaha}$, ii) the input sequence length of LSTM, and iii) the option to use ground-truth bounding boxes or detected boxes with further regression. We tune or verify them on ETH-Bahnhof and Venice-2. The other hyper-parameters, such as P_D, λ_{fa}, λ_{nt} are set to default values as in [63]. We then compare our RMNet to an alternative simple Siamese CNN for metric learning.

FIGURE 8. Some results on the test videos of 2DMOT2015. Object ID is illustrated on the top-left of each box. For better visualization, please use the PDF version.

1) EVALUATION OF $T_{MaxMaha}$

As shown in Figure 5(left), the performance is gradually improved with the increasing of $T_{MaxMaha}$ but saturated after about 14. In fact, too small $T_{MaxMaha}$ could reject hypothesis largely and lead to low recall rate. Increasing $T_{MaxMaha}$ can retain high recall but is not sensitive to the final performance since our method can reject the mismatchings subsequently.

2) EVALUATION OF SEQUENCE LENGTH

To evaluate training sequence length of RMNet, we perform training with different sequence lengths. The sequence length refers to the input number of frames used for LSTM. We show the MOTA metric with different sequence length in Figure 5(right). Increasing length significantly boosts MOTA in the beginning, and saturates after 20 frames. We choose default length as 23 since it provides the highest MOTA. The curve indicates that adding frames is critical for batch multi-object tracking since more temporal information is provided.

3) EVALUATION OF BOUNDING BOX REGRESSION

As mentioned above, our RMNet uses the detected bounding boxes and ground-truth boxes to learn a regressor in training step. To evaluate the effects of bounding box regression in RMNet, we conduct an experiment without the bounding box regression part. We refer DET_R, DET to the default regressor option, no regressor with detected boxes. Figure 6 shows the comparison. In general, training with the regression part improves the performance slightly. The regression part in RMNet provides extra supervised information for training and slightly shifts the detected boxes to the right direction.

TABLE 4. Performance comparison between our tracking method and others methods for the KITTI PEDESTRIAN dataset.

Method	IDF1	Rcll	Prcn	FAR	FP	FN	IDSW	MOTA	MOTP
CEM [7]	50.51	36.73	80.82	18.16	2020	14658	96	27.54	**68.48**
NOMT-HM [65]	50.89	37.30	80.09	19.35	2153	14559	**73**	27.49	67.99
Ours	55.64	43.69	76.58	27.92	3106	13085	321	**28.67**	67.80

4) IMPACT OF TRAINING WITH DETECTION

To evaluate the effects of training with detection instead of ground truths, we present another two experiments. The first one uses ground-truth boxes for training and the other one uses detected boxes. Figure 6 shows the comparison where GT means no regressor with ground-truth boxes. In general, training with detected boxes consistently outperforms the one with ground-truth boxes. We argue that training with GT box may suffer from the gap between training and testing since only detected boxes are available in the test phase.

5) A COMPRISON WITH SIAMESE CNN FOR METRIC LEARNING

To verify the effect of the proposed sequence-wise RMNet model, we also perform our algorithm with a pair-wise deep metric, which is implemented by a simple Siamese convolutional neural networks [64]. This pair-wise deep metric only compares the last sample with current detection and produces the similarity. The result is presented as Table. 5, our RMNet significantly boosts the performance of the tracking method according to MOTA, IDF1, etc.

TABLE 5. Performance comparison with different appearance models.

Method	IDF1(↑)	IDSW(↓)	MOTA(↑)	MOTP(↑)
Pair-wise	55.2	40	34.9	72.5
RMNet	64.2	6	50.7	79.0

6) SPEED EVALUATION

We present the run-time speed evaluation in the last column of Table 1. Unlike other trackers that provide running time for only tracking process without feature extraction and other operations, we also evaluate the whole procedure including feature extraction, RMNet updating and BMH execution. We get 16.9 Hz (frames per second) for the whole procedure and more than 300 Hz for only the tracking process (BMH).

7) VISUALIZATION

In Figure 7, we show an instance and illustrate how the RMNet and appearance score react. In this segment, the object of ID 228 in frame 67 has been overlapped by other objects several times. Frame 78 and 104 of Figure 7(c) present an occlusion example and an inaccurate detection respectively. Due to the frequent severe occlusion and inaccurate detection, missing or mismatches could often happen and influence appearance model. As the Figure 7(a) shows, the proposed RMNet still can produce a high response when meeting correct detection again such as the re-tracking in frame 124, which indicates that it maintains high reliability during complicated situation. Figure 7(b) is the corresponding S_{app} (Eq. (9)). Occlusion or inaccurate detection decreases S_{app} significantly while accurate matching or re-tracking boosts the S_{app}. Figure 8 shows some tracking results on test videos.

VII. CONCLUSION

In this paper, we propose a RMNet for recursive similarity metric learning and a batch multiple hypothesis (BMH) strategy based tracker for multi-object tracking. The RMNet is trained on sequences with mixed positive and negative bounding box with outputs of a binary similarity and a bbox regression. Experiments show its effectiveness in dealing with mis-matches due to its long-short term memory units. The BMH is a tree based structure, and generates hypotheses with a dual-threshold scheme. We also introduce a re-find reward in BMH to handle missing detections. We conduct extensive experiments on the 2DMOT2015 dataset and obtain the state-of-the-art ID measures.

ACKNOWLEDGMENT

(Longtao Chen and Xiaojiang Peng contributed equally to this work.)

REFERENCES

[1] C.-W. Lu, C.-Y. Lin, C.-Y. Hsu, M.-F. Weng, L.-W. Kang, and H.-Y. M. Liao, "Identification and tracking of players in sport videos," in *Proc. Int. Conf. Internet Multimedia Comput. Service*, New York, NY, USA, 2013, pp. 113–116, doi: 10.1145/2499788.2499842.

[2] P. Nillius, J. Sullivan, and S. Carlsson, "Multi-target tracking—Linking identities using Bayesian network inference," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 2006, pp. 2187–2194.

[3] J. Xing, H. Ai, L. Liu, and S. Lao, "Multiple player tracking in sports video: A dual-mode two-way Bayesian inference approach with progressive observation modeling," *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1652–1667, Jun. 2011.

[4] A. Ess, K. Schindler, B. Leibe, and L. Van Gool, "Improved multi-person tracking with active occlusion handling," in *ICRA Workshop on People Detection and Tracking*, vol. 2. Citeseer, 2009.

[5] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, Jun. 1989.

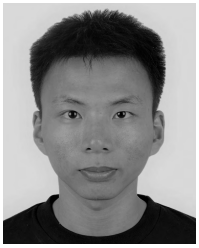
[6] A. Ess, B. Leibe, K. Schindler, and L. Van Gool, "A mobile vision system for robust multi-person tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2008, pp. 1–8.

[7] 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.

[8] A. Milan and L. Leal-Taixé, K. Schindler, and I. Reid, "Joint tracking and segmentation of multiple targets," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5397–5406.

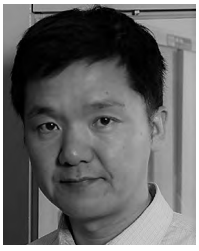
[9] S. Wang and C. C. Fowlkes, "Learning optimal parameters for multi-target tracking with contextual interactions," *Int. J. Comput. Vis.*, vol. 122, no. 3, pp. 484–501, May 2017.
[10] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, no. 1, Jun. 2005, pp. 886–893.
[11] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, A. Leonardis, H. Bischof, and A. Pinz, Eds. Berlin, Germany: Springer, 2006, pp. 428–441.
[12] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[13] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face recognition with local binary patterns," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, T. Pajdla and J. Matas, Eds. Berlin, Germany: Springer, 2004, pp. 469–481.
[14] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining Knowl. Discovery*, vol. 2, no. 2, pp. 121–167, Jun. 1998.
[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[17] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.
[18] X. Liang, X. Shen, D. Xiang, J. Feng, and L. L. S. Yan, "Semantic object parsing with local-global long short-term memory," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3185–3193.
[19] L. Wang, W. Ouyang, X. Wang, and H. Lu, "STCT: Sequentially training convolutional networks for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1373–1381.
[20] B. Wang *et al.*, "Joint learning of convolutional neural networks and temporally constrained metrics for tracklet association," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun./Jul. 2016, pp. 386–393.
[21] L. Leal-Taixé, C. Canton-Ferrer, and K. Schindler, "Learning by tracking: Siamese CNN for robust target association," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun./Jul. 2016, pp. 418–425.
[22] H. Mahgoub, K. Mostafa, K. T. Wassif, and I. Farag, "Multi-target tracking using hierarchical convolutional features and motion cues," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 11, pp. 1–6, 2017.
[23] A. Milan, S. H. Rezatofighi, A. Dick, I. Reid, and K. Schindler, "Online multi-target tracking using recurrent neural networks," in *Proc. Assoc. Advance Artif. Intell. (AAAI)*, Feb. 2017, pp. 4225–4232.
[24] M. Kojima, H. Kameda, S. Tsujimichi, and Y. Kosuge, "A study of target tracking using track-oriented multiple hypothesis tracking," in *Proc. SICE Annu. Conf. Int. Session Papers*, Jul. 1998, pp. 933–938.
[25] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler. (Apr. 2015). "Motchallenge 2015: Towards a benchmark for multi-target tracking." [Online]. Available: https://arxiv.org/abs/1504.01942
[26] J. Ferryman and A. Shahrokni, "Pets2009: Dataset and challenge," in *Proc. 12nd IEEE Int. Workshop Perform. Eval. Tracking Surveill.*, Dec. 2009, pp. 1–6.
[27] M. Andriluka, S. Roth, and B. Schiele, "People-tracking-by-detection and people-detection-by-tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
[28] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.
[29] J. H. Yoon, M.-H. Yang, J. Lim, and K.-J. Yoon, "Bayesian multi-object tracking using motion context from multiple objects," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2015, pp. 33–40.
[30] J. H. Yoon, C.-R. Lee, M.-H. Yang, and K.-J. Yoon, "Online multi-object tracking via structural constraint event aggregation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1392–1400.
[31] S.-H. Bae and K.-J. Yoon, "Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1218–1225.
[32] S. Huang, S. Jiang, and X. Zhu, "Multi-object tracking via discriminative appearance modeling," *Comput. Vis. Image Understand.*, vol. 153, pp. 77–87, Dec. 2016, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1077314216300790
[33] C.-H. Kuo and R. Nevatia, "How does person identity recognition help multi-person tracking?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 1217–1224.
[34] 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.
[35] K. Okuma, A. Taleghani, N. de Freitas, J. J. Little, and D. G. Lowe, "A boosted particle filter: Multitarget detection and tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, T. Pajdla and J. Matas, Eds. Berlin, Germany: Springer, 2004, pp. 28–39.
[36] Z. Khan, T. Balch, and F. Dellaert, "MCMC-based particle filtering for tracking a variable number of interacting targets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 11, pp. 1805–1819, Nov. 2005.
[37] S. Oh, S. Russell, and S. Sastry, "Markov chain Monte Carlo data association for multi-target tracking," *IEEE Trans. Autom. Control*, vol. 54, no. 3, pp. 481–497, Mar. 2009.
[38] S. Kim, S. Kwak, J. Feyereisl, and B. Han, "Online multi-target tracking by large margin structured learning," in *Proc. Asian Conf. Comput. Vis.*, K. M. Lee, Y. Matsushita, J. M. Rehg, and Z. Hu, Eds. Berlin, Germany: Springer, 2013, pp. 98–111.
[39] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, "Online multiperson tracking-by-detection from a single, uncalibrated camera," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1820–1833, Sep. 2011.
[40] A. Andriyenko, K. Schindler, and S. Roth, "Discrete-continuous optimization for multi-target tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 1926–1933.
[41] C. Dicle, O. I. Camps, and M. Sznaier, "The way they move: Tracking multiple targets with similar appearance," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2304–2311.
[42] 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.
[43] A. Milan, K. Schindler, and S. Roth, "Detection- and trajectory-level exclusion in multiple object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 3682–3689.
[44] L. Wen, Z. Lei, S. Lyu, S. Z. Li, and M.-H. Yang, "Exploiting hierarchical dense structures on hypergraphs for multi-object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 1983–1996, Oct. 2016.
[45] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes, "Globally-optimal greedy algorithms for tracking a variable number of objects," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 1201–1208.
[46] 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. (CVPR)*, Jun. 2008, pp. 1–8.
[47] J. Munkres, "Algorithms for the assignment and transportation problems," *J. Soc. Ind. Appl. Math.*, vol. 5, no. 1, pp. 32–38, 1957.
[48] D. B. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Autom. Control*, vol. 24, no. 6, pp. 843–854, Dec. 1979.
[49] Z. He, X. Li, X. You, D. Tao, and Y. Y. Tang, "Connected component model for multi-object tracking," *IEEE Trans. Image Process.*, vol. 25, no. 8, pp. 3698–3711, Aug. 2016.
[50] B. Wang, G. Wang, K. L. Chan, and L. Wang, "Tracklet association with online target-specific metric learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1234–1241.
[51] L. Chen, H. Ai, C. Shang, Z. Zhuang, and B. Bai, "Online multi-object tracking with convolutional neural networks," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 645–649.
[52] 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.
[53] A. Sadeghian, A. Alahi, and S. Savarese, "Tracking the untrackable: Learning to track multiple cues with long-term dependencies," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 300–311.
[54] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 580–587.
[55] N. McLaughlin, J. M. D. Rincon, and P. Miller, "Enhancing linear programming with motion modeling for multi-target tracking," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Jan. 2015, pp. 71–77.

[56] L. Leal-Taixé, M. Fenzi, A. Kuznetsova, B. Rosenhahn, and S. Savarese, "Learning an image-based motion context for multiple people tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 3542–3549.
[57] S. H. Rezatofighi, A. Milan, Z. Zhang, Q. Shi, A. Dick, and I. Reid, "Joint probabilistic data association revisited," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3047–3055.
[58] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1532–1545, Aug. 2014.
[59] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *EURASIP J. Image Video Process.*, vol. 2008, p. 246309, May 2008.
[60] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *Proc. Eur. Conf. Comput. Vis.*, Cham, Switzerland, 2016, pp. 17–35.
[61] H. Yang, S. Qu, C. Chen, and B. Yang, "Multiple objects tracking with improved sparse representation and rank based dynamic estimation," *IEEE Access*, vol. 6, pp. 42264–42278, 2018.
[62] 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.
[63] I. J. Cox and S. L. Hingorani, "An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 2, pp. 138–150, Feb. 1996.
[64] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jun. 2005, pp. 539–546.



LONGTAO CHEN received the B.E. degree in computer science and technology from the Nanjing University of Science and Technology, Nanjing, Jiangsu, China, where he is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering. From 2016 to 2017, he was a Trainee with Idiap Institute, Switzerland. His current research interests include multi-object tracking and pedestrian tracking.

XIAOJIANG PENG received the Ph.D. degree from the School of Information Science and Technology, Southwest Jiaotong University, in 2014. He was a Postdoctoral Researcher with the LEAR Team, INRIA, France, working with Prof. C. Schmid, from 2015 to 2016, and a Postdoctoral Researcher with the Idiap Institute, Switzerland, from 2016 to 2017. He is currently an Associate Professor with the Multimedia Lab, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China. His research interests include the areas of action recognition and detection, face recognition, and deep learning. He serves as a Reviewer for the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON MULTIMEDIA, *Image and Vision Computing*, *Machine Vision and Applications*, the IEEE SIGNAL PROCESSING LETTERS, *Multimedia Tools and Applications*, *Neurocomputing*, *IET Computer Vision*, FG, and so on.



MINGWU REN received the Ph.D. degree in pattern recognition and intelligent system from the Nanjing University of Science and Technology, Nanjing, Jiangsu, China, in 2001, where he is currently a Professor with the School of Computer Science and Engineering. His current research interests include computer vision, image processing, and pattern recognition.