IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 16, 2019, accepted February 19, 2019, date of publication February 25, 2019, date of current version March 13, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2901520

Online Multi-Object Tracking Based on Feature Representation and Bayesian Filtering Within a Deep Learning Architecture

JUN XIANG, GUOSHUAI ZHANG, AND JIANHUA HOU

College of Electronics and Information Engineering, South-Central University for Nationalities, Wuhan 430074, China

Corresponding author: Jianhua Hou (zil@scuec.edu.cn)

This work was supported in part by the Project of the National Natural Science Foundation of China under Grant 61671484 and Grant 61701548, in part by the Hubei Natural Science Foundation under Grant 2018CFB503, and in part by the Fundamental Research Funds for the Central Universities, South-Central University for Nationalities, under Grant CZZ18001 and Grant CZY18046.

ABSTRACT In detection-based multi-object tracking (MOT), one challenging problem is to design a robust affinity model for data association. Moreover, since these approaches entirely rely on detection responses to locate targets, a strategy should be taken to deal with a detector's defect. In this paper, we propose a robust online MOT tracking method that can handle these two issues effectively. We first present a novel affinity model by jointly learning more powerful feature representation and distance metric within a deep architecture. Specifically, we design a convolutional neural network to extract appearance cue tailored toward person Re-ID and a long short-term memory network to extract motion cue to encode dynamics of targets. Both the cues are then combined with a triplet loss function, which performs end-to-end deep metric learning to encode dependences across both cues automatically and thus generates fused features in embedding space to distinguish targets. To overcome the detector's limitation, a trajectory estimation strategy is presented. We design a recurrent neural network-based Bayesian filtering module, which takes a hidden state of the above-mentioned LSTM network as an input and performs recursive prediction and update for explicitly estimating targets state. In this way, we can reconstruct trajectories by filling the gaps where no detections are present or adjusting the exact locations of trajectory where detections are imprecise. The experiments on the challenging MOT 2015 and 2016 datasets show very competitive results when comparing our method with the recent state-of-the-art batch and online tracking approaches. We achieve top one in terms of multiple objects tracking accuracy and multiple objects tracking precision among online methods on the MOT2016 dataset.

INDEX TERMS Multi-object tracking, deep learning, data association, trajectory reconstruction.

I. INTRODUCTION

Multi-object tracking (MOT) aims at locating multiple objects, maintaining their identities, and yielding individual trajectories given an input video. MOT is of significant relevance for various applications, such as video surveillance, human behavior analysis, autonomous driving and robot navigations. Despite substantial progress in recent years, multi-object tracking remains very challenging when dealing with large appearance variation, high motion complexity, interactions and occlusions among multiple objects [1].

To address this challenge, most current approaches follow tracking-by-detection framework, where object detectors like [2]–[4] provide potential locations of the objects of interest in the form of bounding boxes, i.e. detection responses. The task of multi-object tracking is then cast as a data association problem, where detection responses are assigned different unique IDs corresponding to different targets, based on shape, motion, and appearance cues [5]. To achieve this, an affinity model is required to estimate the linking probability (also called assignment cost) between detections and targets, followed by an association optimization strategy that determines which of the targets should be linked considering their affinity measurements. The main motivation in this paper is to develop a robust affinity model.

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

2169-3536 © 2019 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

Feature representation is crucial in affinity model designing. In the past decades, different kinds of features have been developed to build a robust appearance model, including color histograms [6]–[8] the Histogram of Gradients (HOG) [9], [10], covariance matrix [11], [12] etc. To produce more accurate affinity scores, motion cues like velocity or position are also utilized to encode dynamics of targets and combined with the target appearance [8], [12]–[15]. While appearance and motion cues are complementary, each of the two is usually treated as separate learning task and the final affinity score is a simple linear combination without metric learning mechanism in many previous works. When faced with MOTChallenge benchmarks [16], [17], where more challenging scene is presented such like great appearance deformation, very low illumination and severe occlusions, the discriminability of the above representation learning methods is often limited due to their shallow representation models. In this case, it is essential to learn more powerful feature representation so that the affinity model is robust to the variation between intra-target, yet remains discriminative for the inter-target variation.

Obviously, if the detector is flawless at finding all targets, the multi-object tracking problem can be solved solely by data association [14], [18], [19]. However, object detectors are not perfect: missed detections, false alarms, and inaccurate responses are still common in challenging real-world environments. For instance, missed detections(i.e., false negatives) caused mainly by severe occlusion will lead to track fragments during the data association stage. The presence of projected shadows, clutter, or partial occlusion usually produce imprecise detection responses, which will result in inaccurate target localization and size. To remedy this, trajectory estimation has been utilized to reconstruct the entire trajectory of each target by filling in the gaps where no detections are present, or to adjust the exact course of a trajectory which tends to deviate due to imprecise target localization [14], [18]. Although interpolation has been used in [12], [20], and [21], as a straightforward method, it cannot handle such situations where targets move unpredictably. Some previous approaches employ delicate techniques for this purpose, e.g., particle filtering based prediction [22] and trajectory extension [23]. However, motion models in these approaches are still relatively simple and do not consider the interactions between targets. It should be noted that a reasonable strategy is to estimate the state of trajectory by modeling the temporal dynamic of targets [8], [24], but explicitly reasoning the state of each target is relatively less investigated in the recent literature of MOT despite its importance.

Over the last few years, the computer vision community has gone through a revolution fueled by deep learning. As deep neural networks (DNNs) can learn rich representations, a current trend of MOT is to directly extract hierarchical features from raw images by DNNs and then to build an affinity model [25]–[29]. Another paradigm that has been used in conjunction with discriminative representation is metric learning. In this setting, a distance metric between measurements in an embedding space is learned from training data to address the variability in object appearance. For instance, methods in [30] and [31] have been attempted to jointly learn the deep features and temporally constrained metrics in a unified convolutional neural networks (CNNs). Although some significant progress has been achieved, there are only a few deep learning approaches to multi-object tracking [31] compared with other computer vision tasks such like object detection and recognition [24]. Furthermore, the problem of trajectory estimation is not taken into account in these deep learning methods.

Motivated by the above fact and recent success of deep learning in computer vision, this paper attempts to address two problems in detection based multi-object tracking: (1) affinity model designing for data association and (2) trajectory estimation to deal with detector's defect, both are investigated within a unified deep architecture.

To build a robust affinity model, we propose to learn feature representations based on multiple cue and deep metric learning. Specifically, we employ a CNN to extract appearance cue tailored towards person Re-ID, and a Long Short-Term Memory network (LSTM) to extract motion cue aiming at encoding position information of each target. Both feature cues are then combined with a triplet loss function that performs end-to-end deep metric learning and generates the embedding of the fused appearance and motion features. The proposed CNN+LSTM model together with a triplet loss function can be considered as learning a mapping function that maps each detection into an embedding space, where the difference between detections of the same target is less than that of different targets.

To overcome detector's limitation, a recurrent neural networks (RNN) that is embedded in the above motion model is designed to conduct classical Bayesian filtering for the task of trajectory estimation. The hidden state of the LSTM that encode targets dynamics is fed into the RNN. In the proceeding of data association, a noisy detection may be assigned to a previous target and serves as a new measurement to update prediction of the target state. A detection may also be associated to no target when a new object appears or detector runs wrong. In this situation, no available observation is provided for updating step, and the state estimated by hidden state in prediction stage is used instead as the target trajectory. Intuitively, Bayesian filtering facilitates more accurate localization of targets, and consequently results in more reliable and smoothed trajectories.

To summarize, the main contributions of this paper are as follows:

1) Proposition of a unified deep architecture to address affinity model for data association, and trajectory estimation for better targets' localization.
2) Proposition of a robust affinity model by learning strong feature representations based on multiple cues and deep distance metric.

3) Proposition of a RNN-based Bayesian filtering module to deal with detector's defect by explicitly inferring the location of each target.

The rest of the paper is organized as follows. We first discuss related work in Section II. The architecture and characteristics of our approach are presented in Section III. We describe the implementation details in both training and testing stage in Section IV. The experimental results are presented and discussed in Section V, and conclusions are drawn in Section VI.

II. RELATED WORK

As summarized in [18], the task of detection based multi-object tracking is twofold: data association and trajectory estimation. In this section, we briefly review previous research relevant to our work, with a focus on deep learning based approaches.

A. FEATURE REPRESENTATION

Feature representation is critical for affinity model in data association, and usually depends on the combination of multiple cues, e.g. appearance and motion.

1) APPEARANCE CUE

Most traditional methods adopt weak affinity measures based on appearance model such as spatial affinity, e.g. bounding box overlap or Euclidean distance [32], [33], or simple appearance similarity, e.g. intersection kernel with color histogram [34]. With the recent rise of deep learning, CNNs are exploited to extract hierarchical features to model appearance similarity [25]–[29]. The architecture extensively used in MOT is the Siamese network [26], [30]. Siamese network processes two inputs simultaneously using multiple layers with shared weights and seems useful for the task of comparing two image patches. However, Siamese models trained with verification loss (or binary classification) only answer the question ''How similar are these two detections or patches?'' [35], without taking into account ''where and when these detections originated'' [26]. To alleviate this problem, Leal-Taixé *et al*. [26] train a CNN in a Siamese configuration, and the outputs of CNN are combined with contextual features by a gradient boosting algorithm. A more appropriate viewpoint is to treat MOT as a retrieval or Re-ID problem, and to build appearance model based on identity classification loss (i.e. identity preserving loss). Along this line, Tang *et al*. [28] developed a Siamese ID-Net for person Re-ID, where the appearance learned by deep networks is combined with body pose information.

2) MOTION CUE

In scenarios where objects have similar appearance or small size, motion features can capture the dynamic nature of the scene that is complementary to appearance features, and therefore are usually utilized in conjunction with appearance to predict the target location [10], [14], [24], [27], [30]. Although linear motion models are popular [10], [30], [36] with a priori assumption that targets follow a linear movement with constant velocity across frames, the simple mechanism behind linear model makes it hard to produce more accurate prediction, and might yield unrealistic or unreasonable trajectories due to the complexity of human motion patterns. In [15], more sophisticated linear model has been taken into account to increase the discriminating power for association, where a dynamic motion affinity is considered by modeling the target motion as a sequence of piecewise linear regressions from the available trajectory. Several works proposed non-linear motion models that consider more complex motion dependency between targets [37], [38]. For instance, Yang and Nevatia *et al*. [37] employ a non-linear motion model to handle the situation that targets may move freely. Recently, recurrent neural networks (RNNs) are utilized to model non-linear motion pattern [24], [27]. RNNs, in particular LSTM networks, are very powerful in capturing spatial and temporal dependencies in data sequences by using non-linear transformations and hidden-state memory built into the LSTM cells [39]. Milan *et al*. [24] propose a RNN based network to learn complex motion model under Bayesian filtering framework, and the learned temporal dynamics of targets is utilized to perform state prediction and updating as well as track management. Sadeghian *et al*. [27] present a LSTM model to predict similar motion patterns by considering the past movements of an object and predicting its future trajectory. To encode long-term temporal dependencies, a hierarchical RNN is used to jointly reason on motion, appearance and interaction cues over a temporal window.

Similar to [27], our approach integrates multiple cues into feature description. However, our system differs in several aspects. First, instead of a regular Siamese CNN in [27], we employ a CNN tailored towards person Re-ID to extract appearance cue. Second, we model motion dynamics by predicting target position instead of velocities. Last, we extend LSTM architecture by incorporating Bayesian state estimation for the task of trajectory estimation, which could further improve tracking performance.

3) DEEP METRIC LEARNING

Different from traditional metric learning methods [40], [41] which learn a single linear transformation to project original data points into another feature space, deep metric learning (DML) cast the problem as a constrained optimization problem within deep neural networks and explicitly learn several hierarchical non-linear transformations. In [30], a Siamese CNN pre-trained on the auxiliary data is used to extract appearance features. A loss function is proposed consisting of a common Mahalanobis distance metric and a temporally constrained segment-wise metric. The Siamese CNN and temporally constrained metrics are jointly learned to generate the appearance-based tracklet affinity. In [31], a quadruplet architecture (Quad-CNN) is proposed to learn more sophisticated representations. A bounding box regression loss and a multi-task ranking loss that considers appearance and

motion-aware position between four images are employed to jointly optimize the Quad-CNN end-to-end.

Different from [30] and [31], our method employs CNN and LSTM to learn appearance and motion features separately. Furthermore, a triplet loss is adopted to perform end-to-end deep metric learning and generates the embedding of the fused appearance and motion features.

B. TRAJECTORY ESTIMATION

Most existing tracking techniques entirely rely on detection responses to derive the locations of targets, However, detectors are not perfect: missed detections, false alarms, and inaccurate responses are still common in challenging real-world scenarios. A solution to this problem is trajectory estimation, i.e., to fill in the gaps where no detections are present, or to adjust the exact locations of trajectory where detections are imprecision. In [8], the state (i.e. the location) of each target is explicitly represented, and then estimated directly by minimizing an energy function which is defined in continuous space. To deal with trajectory gaps due to occlusion or detection failures, Tang *et al*. [28] estimate a smoothed trajectory from the detections that belongs to the same cluster in a similar manner in [8], and then fill in the missing detections along the estimated trajectory. Trajectory extension is also employed in [23] to track reliable targets so that missed head or tail parts of tracklets are partially recovered. In [36], particle filtering based strategy is presented, where a novel mutual occlusion reasoning and targets' interactions are considered for more accurate observation likelihood and refining the final trajectories. Bounding box regression is also leveraged for better localization [31]. However, this vision based strategy is limited in the case of heavy occlusions since the observations of an occluded target may drastically decrease even if the estimated location is accurate. More recently, Milan *et al*. [24] present an elegant formulation that employs an end-to-end training for multi-object tracking. In their work, a RNN-based architecture is used for Bayesian state estimation, i.e. trajectory estimation, and a LSTM-based model is designed for data association.

Inspired by [24], we design a neural network to conduct Bayesian inference that is composed of both state prediction and updating. Furthermore, the Bayesian neural network is embedded in LSTM motion model, which not only enables the extended LSTM architecture to capture motion patterns containing identity to distinguish targets, but also allows for explicitly estimating the location of each target for the task of trajectory estimation.

III. MULTIPLE OBJECT TRACKING FRAMEWORK

Given a set of already tracked targets at current time, the task of our tracking framework is to associate each new detection in the next time to a corresponding target. Meanwhile, a trajectory estimation strategy is employed to overcome the detector's limitation and to improve final tracking performance. In this section, we describe the details of components in our approach, including how to build affinity model for data association and how to design Bayesian filtering network for trajectory estimation. The architecture overview of our tracking framework is shown in Fig. 1.

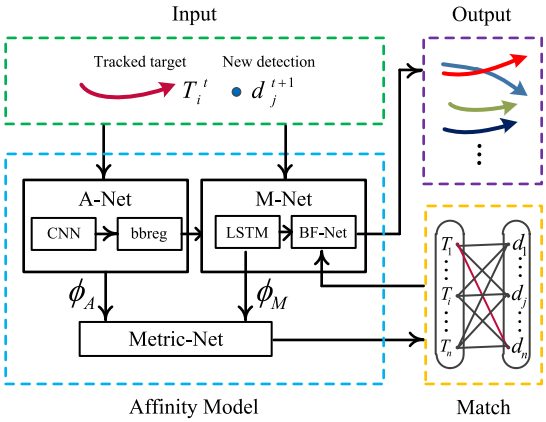


FIGURE 1. Architecture overview. The system consist of two parts. In affinity model (the blue dashed rectangle),the A-Net and LSTM network extract appearance cue and motion cue, respectively. Both cues are combined in Metric-Net with a triplet loss function that performs end-to-end learning of multiple-cue representations and produces the desired embedding features. In matching part (the yellow dashed rectangle), a bipartite graph is constructed by association cost between already tracked targets and new detections, and inference is achieved by Hungarian algorithm. Finally, Bayesian filtering module(BF-Net) embedded in M-Net is employed to refine tracks with or without the associated detections.

A. APPEARANCE MODEL

The motivation of our appearance model is to develop more discriminative and robust representations for visual feature property. To this end, we employ a CNN with the identity classification loss, which is usually used in person Re-ID community.

1) DATA COLLECTION

It is well known that deep architectures require vast amounts of training data in order to avoid overfitting of model. To learn A-Net, we collect training set from two different datasets. Training images are collected firstly from the 2DMOT2015 benchmark training set [16] and 5 sequences in the MOT16 benchmark training set [17]. We also collect person identity examples from the CUHK03 and Market-1501 [42] datasets. For validation set, we use the MOT16-02 and MOT16-11 sequences from the MOT16 training set. Overall a total of 2551 identities are used for training and 123 identities for validating.

2) ARCHITECTURE

We use VGG-16 Net [43] as the base CNN architecture. Specifically, by training VGG-16 to recognize $Y = 2551$ unique identities, the learning can be viewed as a Y-way classification problem. Training images are re-sized to $224 \times 224 \times 3$ and each image I_i corresponds to a ground-truth identity label $l_i \in \{1, 2, \cdots Y\}$. The network is trained using the Softmax loss to estimate the probability of each image being each label by a forward pass.

Since data association relies on accurate object localization, we incorporate bounding-box regression in [31] as an additional objective to learn the A-Net. Specifically, We employ a multi-task loss to jointly learn appearance cue and bounding box regression for better localization. The bounding box regression loss $\mathcal{L}_b$, following the fully connected layer ϕ_{f7}(4096-dimension), is given by

$$\mathcal{L}_b = \sum_{i\in\{u,v,w,h\}} smooth_{L_1}(g_i - p_i) \tag{1}$$

where $g = \{g_u, g_v, g_w, g_h\}$ denotes an offset of ground-truth bounding box, and $p = \{p_u, p_v, p_w, p_h\}$ is a predicted bounding-box regression offset. The L_1 smooth loss function is given by

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

For bounding box regression, we adopt the 4 coordinates parameterizations, specifying the pixel coordinates of the center of box together as well as the box's width and height in pixels. Note that in the test time, we use the fully connected layer ϕ_{f7} as the appearance feature $\phi_A(I_i)$ and the box localizations refined by bounding-box regression as the input for M-Net.

B. MOTION MODEL

Since motion model encodes the dynamics of trajectories that is complementary to appearance cue, it could be exploited to distinguish targets during data association (discrete problem) as well as to predict positions of targets for trajectory estimation (continuous problem for state estimation). This paper proposes a neural network, denoted as M-Net, to learn the temporal dynamics of targets, and attempts to address the above two problems. In this subsection, we explain how to construct a LSTM network with verification loss to extract motion cue for affinity model. The designing of Bayesian filtering network for the task of trajectory estimation is left in subsection III (D). Our M-Net is illustrated in Fig. 2, where the left part represents the LSTM network used to extract motion cues.

1) ARCHITECTURE

The task of motion model is to determine whether a trajectory should be located at a particular position or not. Our LSTM accepts as inputs the positions of trajectory i with length of N frames, denoted as $T_i^t = [p_i^{t-(N-1)}, p_i^{t-(N-2)}, \cdots, p_i^t]$, and produces a H-dimensional output ϕ_i^t. We also pass a position p_j^{t+1} at the next time which we wish to determine whether it corresponds to the true trajectory T_i^t or not, through a fully-connected layer (FC2) that maps it to a H-dimensional vector denoted as ϕ_j^{t+1}. The difference between ϕ_i^t and ϕ_j^{t+1} is passed to another fully connected layer (FC3), followed by a Softmax layer to produce an assignment probability $\mathcal{A}(T_i^t, p_j^{t+1})$ over binary classification. Intuitively, it is reasonable to judge whether position

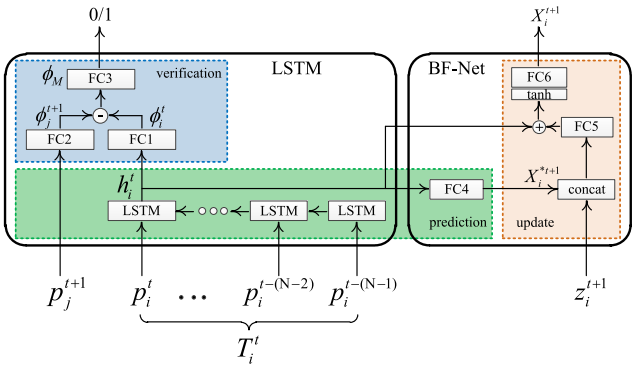


FIGURE 2. Motion model (M-Net) consists of two modules. The first (left part) is LSTM based feature extractor that takes as input a series of position coordinates of a trajectory T_i^t and a position p_j^{t+1}. The second (right part) is Bayesian filtering network denoted by BF-Net. The hidden state h_i^t produced by LSTM and the detection z_i^{t+1} (i.e. measurement) by association inference are fed as input into prediction and update block, respectively.

p_j^{t+1} corresponds to a true trajectory T_i^t by H-dimensional output of layer FC3 that is represented as $\phi_M(T_i^t, p_j^{t+1})$. We define $\phi_M(T_i^t, p_j^{t+1})$ as the final motion feature vector in testing stage.

Given $\mathcal{A}(T_i^t, p_j^{t+1})$, the assignment probability that trajectory T_i^t should be located at p_j^{t+1}, the LSTM based feature extractor is learned by minimizing the binary cross entropy (BCE) loss:

$$\begin{aligned}\mathcal{L}_v = -[&\widehat{\mathcal{A}}(T_i^t, p_j^{t+1}) log \mathcal{A}(T_i^t, p_j^{t+1}) \\ &+ (1 - \widehat{\mathcal{A}}(T_i^t, p_j^{t+1})) log(1 - \mathcal{A}(T_i^t, p_j^{t+1}))]\end{aligned} \tag{3}$$

Here $\hat{\mathcal{A}}(T_i^t, p_j^{t+1})$ is the true association distribution of T_i^t and p_j^{t+1}.

2) DATA COLLECTION

Due to the very tedious and time-consuming task of video annotation, only very limited amount of real data for pedestrian tracking is publicly available. We therefore resort to synthetic data augmentation [24] by sampling from a simple generative trajectory model learned from MOT15 and MOT16. We refer to [24] for more details. There are about 100K trajectories in the collected training set, each of 20 frames in length. The data is divided into mini-batches of 10 samples per batch and normalized to the range $[-0.5, 0.5]$, w.r.t. the image dimensions. As mentioned previously, each sample is a data pair consisting of a trajectory T_i^t (N frames in length) and a position p_j^{t+1}. While positive samples are generated by randomly sampling T_i^t and its true position p_i^{t+1}, negative samples are composed of T_i^t and a position p_j^{t+1} from a different trajectory j.

C. DEEP METRIC LEARNING

Deep metric learning is an efficient strategy to addressing the problems of variability in object appearance and motion [5].

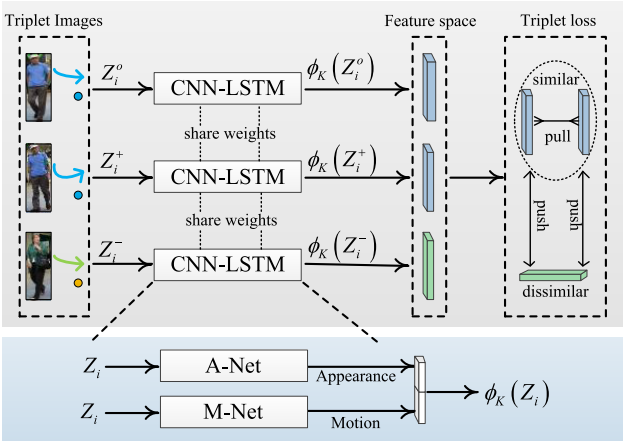


FIGURE 3. Metric-Net training framework. Triplet training images are fed into three-channel CNN-LSTM networks with the shared parameter set. The triplet loss function is used to train the Metric-Net, making the distance between inputs of the same targets is less than that of different targets.

The Metric-Net is designed for that purpose, as illustrated in Fig. 3. The core of this network is a triplet loss function that encodes dependencies across appearance and motion cues automatically. We first pre-trained appearance and motion model separately. Then we train Metric-Net by fine-tuning the weights of each individual component in an end-to-end fashion and fitting appearance and motion features into the triplet loss function. The detail about our deep metric learning is discussed next.

1) DATA COLLECTION

A triplet training example is constituted of an image patch I_i, a trajectory T_i^t and a position p_j^{t+1}. Similar to [44] and [45], we construct a triplet example $Z_i = \langle Z_i^o, Z_i^+, Z_i^- \rangle$ with three items. In anchor $Z_i^o = \langle I_i, T_i^{t_1}, p_i^{t_1+1} \rangle$, I_i, $T_i^{t_1}$ and $p_i^{t_1+1}$ are all from target i. Anchor-positive $Z_i^+ = \langle I_i', T_i^{t_2}, p_i^{t_2+1} \rangle$ is similar to anchor item but with different time stamp. The underlying principle behind our metric learning is to pull together samples from the same class in terms of appearance and motion, while pushing apart those with either different appearance or unreasonable motion state. Consequently, in anchor-negative $Z_i^- = \langle I_j', T_j^{t_3}, p_k^{t_3+1} \rangle$, I_j' and $T_j^{t_3}$ come from a target j that is different from i. Note that in this case, we don't care about whether $p_k^{t_3+1}$ is the real position of $T_j^{t_3}$ or not.

For experiments, we collect triplet examples from MOT15 benchmark training set and 6 sequences of the-MOT16 benchmark training set. The MOT16-02 sequences in the MOT16 training set are used as validation sets. Overall a total of 851 identities are used for training and 54 identities for validating. We generate triplet examples as follows: for each batch of 100 instances, we select 5 persons and generate 20 instances for each person. In each triplet instance, the anchor and anchor-positive are randomly selected from the same identity, and the negative item is also randomly selected, but from the remaining identities.

2) ARCHITECTURE

For Metric-Net training, we design a three-channel appearance-motion model with the shared parameter set. In each channel, one item in a triplet training example Z_i is mapped into a learned feature space to form a (4096 + H)-dimensional vector by concatenating appearance and motion features. A subsequent FC layer is employed for each channel which brings this concatenated feature to a $K = 256$ dimensional embedding space by a triplet loss function, where the embedding feature of Z_i is represented by $\phi_K(Z_i) = \langle \phi_K(Z_i^o), \phi_K(Z_i^+), \phi_K(Z_i^-) \rangle$.The learned embedding space has the desirable property that the distance between $\phi_K(Z_i^o)$ and $\phi_K(Z_i^+)$ is less than the distance between $\phi_K(Z_i^o)$ and $\phi_K(Z_i^-)$ by a predefined margin τ, as described by the following equation:

$$d(\phi_K(Z_i^o), \quad \phi_K(Z_i^+)) - d(\phi_K(Z_i^o), \ \phi_K(Z_i^-)) \leq \tau \qquad (4)$$

where τ is negative.

D. TRAJECTORY ESTIMATION

Another trait of this paper is to propose a deep learning based trajectory estimation strategy to deal with the limitations of detectors, including missed detections and imprecise localizations. We borrow the idea in [24] where a RNN is learned to model the temporal dynamics of targets for Bayesian filtering. To this end, we design a Bayesian filtering network (BF-Net) embedded in the above motion model. The core of our BF-Net is the hidden state outputted by LSTM, which is assumed to capture the complete information necessary for predicting target dynamic model [46] and is given as an input to the BF-Net.

1) BAYESIAN FILTERING

Bayesian approaches are popular for nonlinear/non-Gaussian tracking problems. To obtain dynamic state estimation of targets, this paradigm performs a recursive filtering consisting of essentially two stages: prediction and update [47].

For notational consistency with most previous literature, let x_t and z_t represent the target state and measurement at time step t, respectively. Suppose the probability distribution function (pdf) $p(x_{t-1}|z_{1:t-1})$ at time $t-1$ is available, the prediction stage involves using the system model to obtain the prior pdf of the state at current time t:.

$$p(x_t|z_{1:\,t-1}) = \int p(x_t|x_{t-1}) p(x_{t-1}|z_{1:t-1}) dx_{t-1} \qquad (5)$$

where $p(x_{t+1}|x_t)$ represents the state transition probability and is defined by the system equation under one order Markov assumption.

When a measurement z_t becomes available at time step t, the update stage applies Bayes rule to modify the prediction pdf and obtains the required posterior density of the current state:

$$p(x_t|z_{1:\,t}) \propto p(z_t|x_t) p(x_t|z_{1:\,t-1}) \qquad (6)$$

where $p(z_t|x_t)$ represents observation likelihood defined by measurement model.

Recursive propagation of the posterior density by conducting Equation(5) and (6) forms the basis for Bayesian filtering. Traditionally, two of the most widely used techniques for solving the above equations are Kalman filter [48] and particle filter [49]. In contrast, we leverage a RNN to resolve this optimal Bayesian solution.

2) ARCHITECTURE

When applying Bayesian filtering to multi-object tracking, one is faced with two additional challenges. 1) Before the state update is performed, it is crucial to determine which measurement is associated with which target. In [24], this combinatorial problem of data association is solved by another LSTM network for each frame. Different from [24], the focus of our work is to build a robust affinity model. We achieve the data association inference by Hungarian algorithm, as the yellow dashed rectangle in Fig. 1. 2) The suitable belief state representation as well as the explicit knowledge of the distributions are required to estimate the state of targets [50]. Like many deep learning based approaches, we bypass the need to specify this knowledge explicitly and instead use a highly expressive recurrent neural networks to learn their functions directly from the data. As illustrated in Fig. 2, our BF-Net consists of two blocks corresponding to prediction and update, respectively. In each block, a nonlinear function is learned to that purpose, which will be discussed next.

a: PREDICTION

Given the dynamics and the filtering distribution already estimated at the current time, the prior distribution of state $p(x_{t+1}|z_{1:t})$ for the next time step is derived in the absence of measurements. Specifically, the hidden state (h_t) outputted by LSTM is fed into a fully connected layer (FC4). Assuming this hidden state encodes enough information necessary for predicting motion dynamics of targets, the prior state can be learned as:

$$p(x_{t+1}|z_{1:t}) = \mathcal{F}_1(\theta_p, h_t) \tag{7}$$

where $\mathcal{F}_1(\cdot)$ is the learned prediction function, θ_p represents network's weights. We denote the predicted state as X_i^{*t+1}.

b: UPDATE

Assuming a particular target is associated with a measurement z_i^{t+1} at the current time. We train the network to correct the state distribution based on detection measurement z_i^{t+1}, predicted state X_i^{*t+1} and the hidden state h_t. The posterior state is obtained as:

$$p(x_{t+1}|z_{1:t+1}) = \mathcal{F}_2(\theta_u, h_t, x_i^{*t+1}, z_i^{t+1}) \tag{8}$$

where $\mathcal{F}_2$ is the learned update function, θ_u denotes network's weights. The corrected state of target is outputted by FC6.

Note that when missed detections occur, no measurement is assigned to a particular target. In this case, we cease update operation and use the predicted state as the output of BF-Net.

3) LOSS FUNCTION AND DATA COLLECTION

We train our Bayesian model directly according to Equation (5) and (6), which are predicting the target's position as well as correcting the position state close to the ground truth tracks. To that end, we minimize the mean squared error (MSE) between state predictions and state update and the ground truth:

$$\mathcal{L}_B = \underbrace{\frac{1}{M}\sum ||x^* - \tilde{x}||^2}_{prediction} + \underbrace{\frac{1}{M}\sum ||x - \tilde{x}||^2}_{update} \tag{9}$$

where x^*, x are the predicted value and update value, respectively, $\tilde{x}$ denotes the true state, and M is the number of training samples.

As mentioned in section III (B), for training LSTM based feature extractor, we have generated about 100K trajectories, where each sample is a data pair consisting of a trajectory T_i^t and a position p_j^{t+1}. To learn BF-Net, we randomly select samples from positive data pairs, each pair consisting of a N frames length trajectory and its true position p_i^{t+1}. Furthermore, for each target we generate a noisy detection from the true position as described in [24].

4) TRAINING FOR M-NET

So far we have detailed the whole architecture of M-Net, which can be decomposed into two major components: a LSTM for feature extracting and a BF-Net for trajectory estimation. In fact, the two networks can be trained together as a single one with multiple cost functions provided in Equation(3)and(9). Experimentally, we learn the parameters in M-Net by minimizing $\mathcal{L}_V$ and $\mathcal{L}_B$ alternately. This not only enables the two networks to adapt to each other, but also allows for learning an optimal hidden state representation for Bayesian prediction and update.

IV. IMPLEMENTATION DETAILS

This section describes details of the proposed method in both training and testing stage.

A. TRAINING

It is more desirable to use detection bounding boxes for model training, since the characteristics of detections and ground-truths are different and only detection bounding boxes are available for inference. In training stages we train our models with detection bounding boxes with associated IDs which are generated by using the ground-truth boxes and adding noise to modify the center, width and height of box. To avoid confusion in the training progress, only detections whose visibility are larger than 0.5 are picked. To this end, we borrow the idea in [8] and [14] to design an explicit occlusion reasoning model to compute the visibility for each target. The training proceeds in the following two steps:

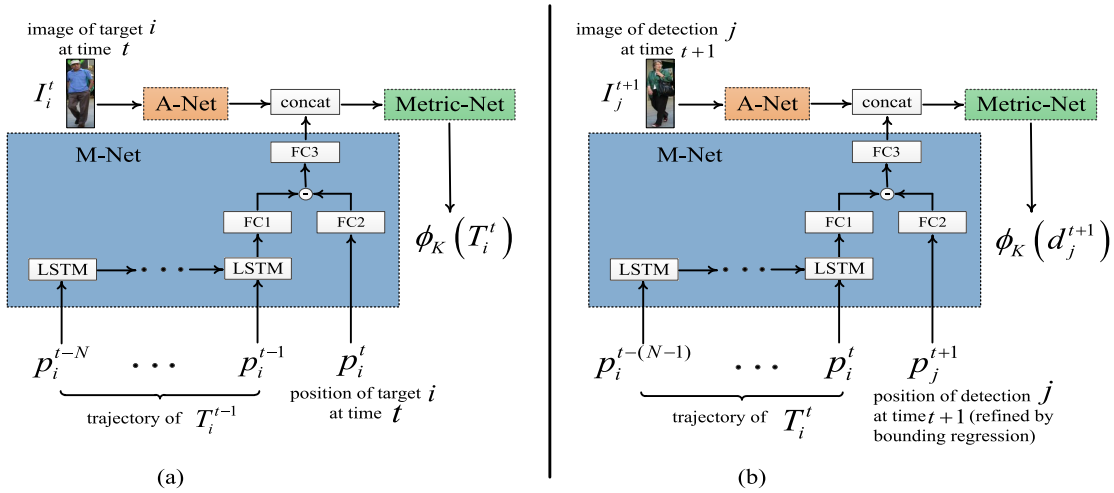


FIGURE 4. The description of our data inputting strategy to compute the association cost between an already tracked target and a new detection.

1) A-Net and M-Net are first pre-trained separately for extracting appearance and motion features, respectively. To learn the A-Net, our VGG model is pre-trained on the ImageNet Classification datasets, and then fine-turned on the MOT and person identity datasets. The learning rate is set initially to 0.0001 and decreased by 10% every 10000 iterations. We set the maximum iteration number to 600000, which is enough to reach convergence. We train M-Net from scratch. The weights are initialized from zero-mean Gaussian distributions with the standard deviation 0.01. The bias terms are set to 0. We use Rmsprop [51] to minimize the loss. The LSTM is trained with one layer and 300 hidden units. And the iteration step is experimentally set to 6 for all data sets. We have also considered using a more complex net size for more representation power but could not achieve reasonable performance. The learning rate is set initially to 0.0003 and decreased by 5% every 20000 iterations. The maximum number of iterations is set to 200000 for convergence. In every 1000 steps, we train M-Net with $\mathcal{L}_V$ loss for the first 500 steps, and with $\mathcal{L}_B$ for the next 500 steps.
2) The metric learning is jointly trained end-to-end with the component A-Net and M-Net. Specifically, we use our A-Net and M-Net as initialization for our metric learning model which makes the training faster and produces better results. In experiments, the parameter τ, the margin of triplet loss function, is experimentally set to -2.

B. TESTING

In testing stage, the main focus is to build a robust affinity model to provide the association cost between an already tracked target and a new detection. We now describe the details of data input strategy and association strategy used in this work.

1) DATA INPUT STRATEGY

For a particular tracked target i at time t, we pass its last image patch (at time t) to A-Net. We also pass its trajectory $T_i^{t-1} = [p_i^{t-N}, p_i^{t-(N-1)}, \cdots p_i^{t-1}]$ and position p_i^t to M-Net, where T_i^{t-1} is fed as input into the LSTM and p_i^t into the FC layer. As illustrated in Fig. 4(a), the output of our affinity model is the embedding feature for target i, denotes as $\phi_K(T_i^t)$.

For a new detection d_j^{t+1}, its image patch at time step $t+1$ is fed into A-Net. As illustrated in Fig. 4(b) and described in sub-section III(B), we pass $T_i^t = [p_i^{t-N-1}, p_i^{t-(N-1)}, \cdots p_i^t]$ and position p_j^{t+1} of d_j^{t+1} to M-Net. In this case, the output of affinity model is $\phi_K(d_j^{t+1})$. The association cost between d_j^{t+1} and target i is defined as the Euclidean distance between $\phi_K(T_i^t)$ and $\phi_K(d_j^{t+1})$:

$$C(T_i^t, d_j^{t+1}) = ||\phi_K(T_i^t) - \phi_K(d_j^{t+1})||^2 \tag{10}$$

Note that to handle noisy detections, we employ bounding box regression in A-Net, and the resulted update localization is given as an input for M-Net.

2) ASSOCIATION STRATEGY

In experiments, we perform two-level association to gradually link detections to longer tracks. In the first round, the tracked targets and new detections in neighboring frames are linked only if the association cost is less than a pre-defined threshold $\eta_1 = 2.5$. The lower the threshold, the less likely detections is associated, leading to fewer ID switches but more fragments. In fact, missed detections, false detections and unassociated detections will inevitably lead to fragments during the first round association. To alleviate this problem, another round of association is taken over those short but reliable tracks in the first round, and fragment tracks are gradually reconnected. To reduce latency, we track targets within a sliding window in the second round, and only appearance cue is employed for association. Specifically, the size

of sliding window is set to 250 frames and overlap between neighboring windows is 50%. In this stage, association costs of fragment pairs are computed by using only appearance embeddings, and the threshold is set to $\eta_2 = 1.5$. Overall, the two-level association strategy will help to improve tracking performance in complex scenes.

Once two-level association is performed, we implement trajectory estimation using BF-Net. The Bayesian filtering not only corrects the tracks in light of new measurements, but also estimates the position of targets in the absence of any detection.

V. EXPERIMENTAL RESULT

In this section, we first describe evaluation metrics. The effect of each component in our method, as well as the result of the ablation study are then analyzed. We demonstrate validity of the proposed method by comparing with several state-of-the-art approaches on the benchmark of MOTChallenge. Finally, we give a further discussion about performance of our method as well as running time.

A. EVALUATION METRICS

We follow the standard MOT2D Benchmark challenge [16] for evaluating multi-object tracking performance. These metrics include: Multiple Object Tracking Accuracy (MOTA↑), Multiple Object Tracking Precision (MOTP↑), Mostly Track targets (MT↑), Mostly Lost targets (ML↓), False Positives (FP↓), False Negatives (FN↓), Fragmentation (FM↓), ID Switches (IDS↓) and finally the number of frames processed in one second (Hz ↑). For items with (↑), higher scores indicate better results;and items with (↓) represent the opposite.

B. EXPERIMENTAL ANALYSIS

In this sub-section, we analyze the performance of each component in our model. We conduct experiments on MOT16 to investigate the validity of appearance cue, motion cue, and metric learning model. 123 person identities collected from MOT16-02 and MOT16-11 are used as test samples. Detections that are considered as true positives for a certain identity are those whose intersection-over-union with the ground truth of the identity are larger than 0.5.

1) VALIDITY OF A-NET

We evaluate our appearance model for identity verification task. Given the true detections for all the test identities, we randomly select 2000 positive pairs assigned to the same identity, and 4000 negative pairs assigned to different identities as our test set. We use the verification accuracy metric, the ratio of correctly classified pairs. The verification result is obtained by comparing the L2 distance between the extracted features and a threshold. The threshold is obtained on a separate validation dataset by maximizing the verification accuracy, which is set to 0.5 in experiment. We also report the verification result of our A-Net in a Siamese architecture manner (denoted as SA-Net), i.e. an additional FC layer on the top of the twin A-Net is employed to model a 2-way classification.

TABLE 1. Validity of A-Net.

Model	Verification Accuracy↑	MOTA↑
A-Net	78.40%	10.40%
SA-Net	84.20%	16.20%

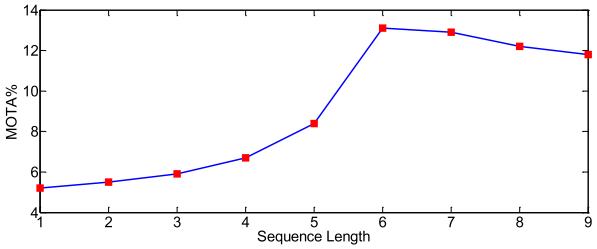


FIGURE 5. Analysis of different sequence length for LSTM mode on the MOT16-02 set.

It can be seen from Table 1 that our A-Net already produces reasonable verification accuracy. The performance is further improved by SA-Net from 78.4% to 84.2%. Moreover, we also report tracking accuracy (MOTA) of the both networks on MOT16-02. While the MOTA result is unsatisfactory due to considering no motion cue, it demonstrates that the A-Net alone can extract meaningful appearance representation for association task. In addition, the result that A-Net achieves a good verification accuracy but a poor MOTA has validated our previous viewpoint that models trained with the verification loss are ''arbitrary'' to some degree when applied in assignment task.

2) VALIDITY OF M-NET

One of the hyper parameters of M-Net is the sequence length N, which is the number of unrolled time steps used while training the LSTM model and enables M-Net being capable to memorize long term dependencies of position cues across time. We investigate the impact of this parameter. Fig. 5 shows the MOTA score under different sequence length for our LSTM model. We can see that increasing the sequence length from 1 to 6 positively impacts the MOTA, and then the performance saturates after 6 frames and the MOTA slightly decreases. It can be explained by the vanishing gradients. Although the architectures of LSTM provide mechanism to deal with the issue of gradient vanishing to some extent, it does not work well in trajectory modeling problem where long-term occlusions occur frequently.

3) VALIDITY OF BF-NET

To demonstrate the functionality of Bayesian filtering in M-Net, we perform experiments on simulated data. Fig. 6 shows an example of tracking result on synthetic data. Four targets in different colors are generated in a rather cluttered environment. The dash and solid lines represent the ground truth trajectories and filtering results, respectively. The filled rectangles are detections. It is observed from Fig. 6 that detections usually deviate far from their ground truth, and rugged curves will be formed accordingly if connecting detections

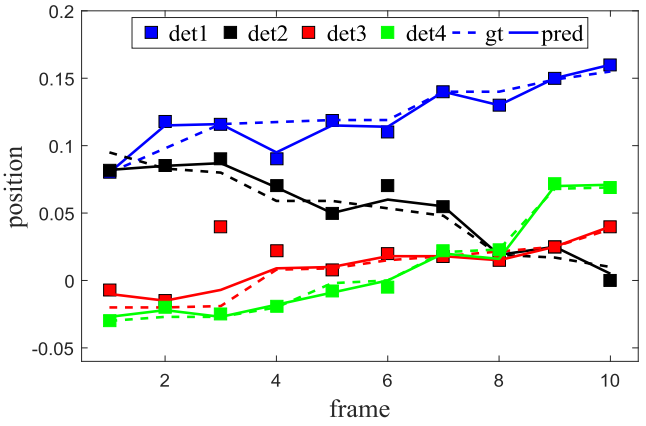


FIGURE 6. Tracking results of Bayesian filtering on synthetic data. The filled rectangles denote the detections, the dash and solid lines represent the ground truth trajectories and filtering results, respectively. And the colors indicate different targets.

of a particular target. In contrast, our Bayesian model can be viewed as performing ''intelligent smoothing'', yielding natural and smooth trajectories (solid lines) that could better fit the ground truth (dash lines).

C. ABLATION STUDY

We investigate the contribution of different components in our framework with detailed tracking metrics on MOT16-02 dataset. Several variants of our algorithm with the same deep architectures are tested. ''T'' and ''V'' mean testing with the triplet loss and verification loss, respectively. ''BR'' and ''BF'' represent testing with bounding box regression and Bayesian filtering, respectively. We denote appearance cue and motion cue by ''A'' and ''M'', respectively. The evaluation results are presented in Table 2 and summarized as follow:

- The appearance cue is the most important one. It can be explained by the fact that representations of people appearance can be learned for varying viewpoint and motion, while less easy to achieve by motion models especially for monocular video sequences due to the complexity of motion. Note that similar conclusions are also reported in [27] and [28].
- The motion cue helps to increase the performance. In highly crowded scenes with clutter and occlusions, our LSTM based motion model can facilitate localization of the targets, while appearance is usually sensitive since the observation likelihood of occluded targets may decrease drastically. In this case, both cues are complementary to make a better performance.
- The triplet loss outperforms the verification loss by a large margin on the available datasets in the terms of MOTA (23% of A + M + T versus 17% of A+M+V). The result also echoes our claim that using triplet loss to optimize the embedding space is more suitable for retrieval or assignment task.
- The bounding-box regression helps to improve tracking metrics. In fact, we employ bounding-box regression to handle noisy localization of detected objects since the motion based association relies on accurate object localization. Compared with A+M+T, the architecture of A + M + T + BR achieves better results in terms of multiple metrics, including MOTA (23.5% versus 23%), MOTP (80.3% versus 74%), FP (147 versus 188), FN (13464 versus 13542) and IDS (48 versus 53).
- The Bayesian filtering network (BF-Net) is learned to model the temporal dynamics of targets, and then is utilized for recursive prediction and update. In this way, the trajectory estimation is implemented. It can be seen from Table 2 that our full model (A+M+T+BR+BF) outperforms all other variant versions in all evaluation metrics, which clearly suggests that each components in our method is helpful for performance gains. Moreover, since BF-Net reduces identity switches significantly (IDS = 33), it yields preferable visual tracking results.

TABLE 2. Analysis of our framework using a different set of components on MOT16-02. (A)Appearance, (M)Motion, (T)Triplet loss, (V)Verification loss, (BR)Bounding box regression and (BF)Bayesian filtering.

Tracker	MOTA↑	MOTP↑	ML↓	MT↑	FP↓	FN↓	IDS↓
A+T	21.90	74.10	**63.00%**	**11.10%**	246	13642	62
M+T	18.50	74.20	65.00%	7.40%	513	13944	73
A+M+T	23.00	74.00	**63.00%**	**11.10%**	188	13542	53
A+M+T+BR	23.50	80.30	**63.00%**	**11.10%**	147	13464	48
A+M+T+BR+BF	**23.60**	**81.70**	**63.00%**	**11.10%**	140	**13458**	**33**
A+V	16.20	74.70	67.00%	5.60%	247	14555	148
M+V	13.10	75.20	67.00%	7.40%	402	14679	410
A+M+V	17.00	74.90	67.00%	5.60%	**94**	14579	123

TABLE 3. Ablation study on the generated 2DMOT2015 validation set. (A)Appearance, (M)Motion, (T)Triplet loss and (BR)Bounding box regression and (BF)Bayesian filtering.

Tracker	MOTA↑	MOTP↑	MT↑	ML↓	FP↓	FN↓	IDS ↓	IDP↑
A+M+T+BR+BF	**34.50**	**76.50**	**13.33%**	61.10%	**188**	**5796**	19	**88.30**
A+M+T+BR	34.30	76.30	**13.33%**	61.10%	197	5805	19	**88.30**
A+M+T	30.10	71.30	10.00%	62.70%	391	5999	**15**	83.50
A+T	25.40	71.90	8.90%	**60.00%**	736	6037	66	71.30
M+T	22.20	72.10	6.60%	61.10%	772	6260	101	62.60

We also conduct ablation study on 2DMOT2015 validation set which includes TUD-Campus, ETH-Bahnhof, ADL-Rundle-8 and KITTI-17. As illustrated in Table 3, conclusions consistent with the above discussion could be drawn.

D. COMPARISON WITH THE STATE OF THE ART

Our full model method(A + M + T + BR + BF), denoted by TripT + BF, are compared with the best published results on the MOT16 test set. The quantitative results presented in Table 4 show that on MOT16, our method achieves the best MOTA, MOTP, ML and FN among all online approaches. Our method even outperforms several offline methods [25], [31], [52] that have access to the whole set of future detections to reason on the data association step, in terms of most metrics except IDS and HZ. The top one tracker in Table 4 is LMP [28], where complex graph-cut association strategy is employed in a batch of several frames and a post trajectory estimation step is used to handle missing detections. In contrast, we adopt a simple linear assignment strategy, which

TABLE 4. Result on the MOT16 test dataset. Best in bold, second best in red, our method is denoted by TripT+BF.

Tracker	Mode	MOTA↑	MOTP↑	MT↑	ML↓	FP↓	FN↓	IDS ↓	HZ↑
LMP [28]	Batch	**48.80**	**79.00**	**18.20%**	**40.10%**	6654	**86245**	481	0.5
Quad-CNN [31]	Batch	44.10	76.40	14.60%	44.90%	6388	94775	745	1.8
MHT-DAM [25]	Batch	42.90	76.60	13.60%	46.90%	5668	97919	499	0.8
LINF1 [53]	Batch	41.00	74.80	11.60%	51.30%	7896	99224	430	1.1
TripT+BF(ours)	Online	48.30	76.70	15.40%	**40.10%**	2706	91047	543	0.5
TripT	Online	48.10	75.50	15.80%	40.20%	2827	91210	563	0.6
AMIR [27]	Online	47.20	75.80	14.00%	41.60%	**2681**	92856	774	1.0
CDA [29]	Online	43.90	74.70	10.70%	44.40%	6450	95175	676	0.5
oICF [52]	Online	43.20	74.30	11.30%	48.50%	6651	96515	**381**	0.4
EAMTT-pub [54]	Online	38.80	75.10	7.90%	49.10%	8114	102452	965	**11.8**
OVBT [55]	Online	38.40	75.40	7.50%	47.30%	11517	99463	1321	0.3

TABLE 5. Result on the 2DMOT2015 test dataset. Best in bold, second best in red, our method is denoted by TripT+BF.

Tracker	Mode	MOTA↑	MOTP↑	MT↑	ML↓	FP↓	FN↓	IDS ↓	HZ↑
Quad-CNN [31]	Batch	33.80	**73.40**	12.90%	36.90%	7898	32061	703	3.7
NOMT [56]	Batch	33.70	71.90	12.20%	44.00%	7762	32547	442	11.5
MHT-DAM [25]	Batch	32.40	71.80	**16.00%**	43.80%	9064	32060	435	0.7
CNNTCM [30]	Batch	29.60	71.80	11.20%	44.0%	7786	34733	712	1.7
LP-SSVM [57]	Batch	25.20	71.70	5.80%	53.00%	8369	36932	646	41.3
SiameseCNN [26]	Batch	29.00	71.20	8.50%	48.40%	**5160**	37798	639	52.8
LINF1 [53]	Batch	24.50	71.30	5.50%	64.60%	5864	40206	**298**	7.5
JPDAm [58]	Batch	23.80	68.20	5.00%	58.1%	6373	40084	365	32.6
AMIR [27]	Online	**37.60**	71.70	15.8%	**26.80%**	7933	**29397**	1026	1.9
TripT+BF(ours)	Online	37.10	72.50	12.60%	39.70%	8305	29732	580	1.0
TripT	Online	35.70	71.70	11.10%	39.80%	8729	30152	655	1.14
TDAM [59]	Online	33.00	72.80	13.30%	39.10%	10064	30617	464	5.9
CDA [29]	Online	32.80	70.70	9.70%	42.20%	4983	35690	614	2.3
MDP [60]	Online	30.30	71.30	13.00%	38.40%	9717	32422	680	1.1
SCEA [61]	Online	29.10	71.10	8.90%	47.30%	6060	36912	604	6.8
RNNLSTM [24]	Online	19.00	71.00	5.50%	45.60%	11578	36706	1490	**165**

is local optimal and whose goal is to match the tracks with detections at a time frame. We speculate that the ranking first performance of LMP is partially attributed to the more sophisticated and delicate optimization strategy, which can be utilized in our future work to obtain better tracking results.

We also present performance of TripT+BF on 2DMOT2015 dataset in the MOTChallenge benchmark. The quantitative results are presented in Table 5, which shows that TripT+BF is very competitive with the state-of-the-art methods. Our proposed method not only achieves the second best tracking accuracy in terms of MOTA, but outperforms all offline methods in Table 5.

E. DISCUSSION

First, we give an analysis between our method and some relevant approaches. AMIR [27] is the most relevant approach to ours where appearance, motion and interaction cues are merged for affinity model and Hungarian algorithm is used for association. On MOT16 test sets our algorithm outperforms AMIR in most evaluation metrics in Table 4. Particularly, we obtain better MOTA (48.3 versus 47.2), MOTP (76.7 versus 75.8), MT(15.4% versus 14%) and IDS (543 versus 774). This means that our method can track more targets (MT) with higher tracking accuracy (MOTA), tracking precision (MOTP), as well as less number of identity switches (IDS), i.e., the performance of our association is more robust. We attribute this excellent property to the proposed framework of learning feature representation and distance metric jointly, which could discriminate targets effectively in crowded scene where occlusions occur frequently, especially in MOT16 dataset. Similar conclusions can also be found in Table 5 on MOT15 test sets, where we have comparable tracking accuracy (MOTA) with better tracking precision (MOTP), as well as less number of identity switches (IDS).

Another two relevant approaches are Quad-CNN [31] and CNNTCM [30], in which the deep metric learning are also adopted in MOT framework. Compared with Quad-CNN [31] that employed the quadruplet loss and bounding box regression, our method obtains an improvement in MOTA (37.1% versus 33.8% in MOT15, and 48.3% versus 44.1% in MOT16). Our MOTA outperforms CNNTCM by a relative large margin(37.1% versus 29.6% in MOT15). The underlying reasons for this improvement are twofold. First, we design a triplet loss to combine appearance and motion cues. Second, instead of CNN used in Quad-CNN [31] and CNNTCM [30], LSTM networks are utilized in our method to encode targets dynamics, which are particular helpful to capture spatial and temporal dependencies in data sequences.

Finally, we investigate the impact of Bayesian filtering network on tracking performance. We compare TripT+BF with its variant denoted as TripT (i.e. without BF-Net). Table 4 and Table 5 have shown that by adding BF-Net for trajectory

estimation, evident performance gains are achieved in most metrics, especially in MOAT (37.1% versus35.7% in MOT15, and 48.3% versus 48.1% in MOT16), MOTP (72.5% versus 71.7% in MOT15, and 76.7% versus 75.5% in MOT16), IDS (580 versus 655 in MOT15, and 543 versus 563 in MOT16). MOTA, MOTP and IDS are the three metrics that most directly depict the quality of tracking and association [62]. In realistic scenarios, lower IDS number often implies better capability to handle occlusion, which is a desirable property in online multiple object tracking. The above consistent gains on both MOTChallenge test sets have demonstrated that by using BF-Net, the proposed trajectory estimation strategy can deal with detector's defect and indeed improve tracking performance.

F. RUNNING TIME

We implemented our framework in TensorFlow on a server with a 2.40GHz CPU and a single GTX 1080Ti GPU. The overall tracking speed of the proposed methods on MOT15 and MOT16 test sequence is 1.0 and 0.5 HZ respectively, excluding the detection step. The results are shown in Table 4 and Table 5, we can see that the proposed method is at least comparable with the state of the art in running time. Note that speed-up can be achieved by further optimization of the codes.

VI. CONCLUSION

In this work, we have proposed a robust multi-object tracking method based on a novel affinity model for data association and a trajectory estimation strategy to deal with detector's defect, both are investigated within a unified deep architecture. To learn more discriminative feature representation, appearance cue and motion cue are extracted separately and are then fused with a triplet loss function in an end-to-end deep metric learning manner. To overcome detector's limitation, we design a RNN based Bayesian network, which can be utilized to perform classical Bayesian filtering for explicitly estimating targets state and thus for trajectory estimation.

Experiments in the challenging MOT benchmark have proved the effect and usefulness of the proposed method. Since we have used a simple linear program algorithm for association, more delicate and effective optimization strategy would be beneficial to further improve the tracking performance.

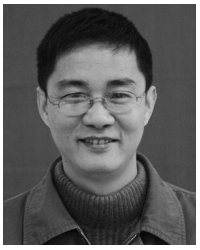
REFERENCES

[1] W. Luo *et al.* (2014). "Multiple object tracking: A literature review." [Online]. Available: https://arxiv.org/abs/1409.7618
[2] N. Dalal and B. Triggs , "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 886–893.
[3] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.
[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
[5] P. Emami *et al.* (2018). "Machine learning methods for solving assignment problems in multi-target tracking." [Online]. Available: https://arxiv.org/abs/1802.06897
[6] D. Mitzel and B. Leibe, "Real-time multi-person tracking with detector assisted structure propagation," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Nov. 2011, pp. 974–981.
[7] H. Izadinia, I. Saleemi, W. Li, and M. Shah, *(MP) 2 T: Multiple People Multiple Parts Tracker*. Berlin, Germany: Springer, 2012.
[8] A. Milan, S. Roth, and K. Schindler, *Continuous Energy Minimization for Multitarget Tracking*. Washington, DC, USA: IEEE Computer Society, 2014.
[9] W. Choi and S. Savarese, "A unified framework for multi-target tracking and collective activity recognition," in *Proc. Eur. Conf. Comput. Vis.*, vol. 7575, no. 1, pp. 215–230, 2012.
[10] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, "Robust tracking-by-detection using a detector confidence particle filter," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 1515–1522.
[11] C.-H. Kuo, C. Huang, and R. Nevatia, "Multi-target tracking by on-line learned discriminative appearance models," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 685–692.
[12] B. Yang and R. Nevatia, "Multi-target tracking by online learning a crf model of appearance and motion patterns," *Int. J. Comput. Vis.*, vol. 107, no. 2, pp. 203–217, 2014.
[13] C. Huang, B. Wu, and R. Nevatia, "Robust object tracking by hierarchical association of detection responses," in *Proc. Eur. Conf. Comput. Vis.*, vol. 5303, Oct. 2008, pp. 788–801.
[14] J. Xiang, N. Sang, J. Hou, R. Huang, and C. Gao, "Multitarget tracking using hough forest random field," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 11, pp. 2028–2042, Nov. 2016.
[15] H. Yang, J. Wen, X.-J. Wu, L. He, and S. G. Mumtaz, "An efficient edge artificial intelligence multi-pedestrian tracking method with rank constraint," *IEEE Trans. Ind. Informat.*, to be published.
[16] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler. (2015). "Motchallenge 2015: Towards a benchmark for multi-target tracking." [Online]. Available: https://arxiv.org/abs/1504.01942
[17] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler. (2016). "Mot16: A benchmark for multi-object tracking." [Online]. Available: https://arxiv.org/abs/1603.00831
[18] A. Milan, "Energy minimization for multiple object tracking," Ph.D. dissertation, Idiap Res. Inst., Techn. Univ. Darmstadt, Darmstadt, Germany, 2014.
[19] A. Heili, A. López-Méndez, and J.-M. Odobez, "Exploiting long-term connectivity and visual motion in CRF-based multi-person tracking," *IEEE Trans. Image Process.*, vol. 23, no. 7, pp. 3040–3056, Jul. 2014.
[20] B. Yang and R. Nevatia, "An online learned CRF model for multi-target tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2034–2041.
[21] A. Heili, C. Chen, and J.-M. Odobez, "Detection-based multi-human tracking using a CRF model," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Nov. 2011, pp. 1673–1680.
[22] 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.
[23] B. Yang and R. Nevatia, *Online Learned Discriminative Part-Based Appearance Models for Multi-human Tracking*. Berlin, Germany: Springer, 2012.
[24] A. Milan, S. H. Rezatofighi, A. Dick, I. Reid, and K. Schindler, "Online multi-target tracking using recurrent neural networks," in *Proc. AAAI*, 2017, pp. 4225–4232.
[25] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg, "Multiple hypothesis tracking revisited," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4696–4704.
[26] 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*, Jun./Jul. 2016, pp. 33–40.
[27] A. Sadeghian, A. Alahi, and S. Savarese. (2017). "Tracking the untrackable: Learning to track multiple cues with long-term dependencies." [Online]. Available: https://arxiv.org/abs/1701.01909
[28] S. Tang, M. Andriluka, B. Andres, and B. Schiele, "Multiple people tracking by lifted multicut and person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 3701–3710.
[29] 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.

[30] 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*, Jun./Jul. 2016, pp. 386–393.
[31] J. Son, M. Baek, M. Cho, and B. Han, "Multi-object tracking with quadruplet convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 3786–3795.
[32] 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.
[33] 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.
[34] A. R. Zamir, A. Dehghan, and M. Shah, "GMCP-tracker: Global multi-object tracking using generalized minimum clique graphs," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, vol. 7573. Berlin, Germany: Springer, 2012, pp. 343–356.
[35] A. Hermans, L. Beyer, and B. Leibe. (2017). "In defense of the triplet loss for person re-identification." [Online]. Available: https://arxiv.org/abs/1703.07737
[36] J. Xiang, N. Sang, J. Hou, R. Huang, and C. Gao, "Hough forest-based association framework with occlusion handling for multi-target tracking," *IEEE Signal Process. Lett.*, vol. 23, no. 2, pp. 257–261, Feb. 2016.
[37] B. Yang and R. Nevatia, "Multi-target tracking by online learning of non-linear motion patterns and robust appearance models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1918–1925.
[38] 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.
[39] H. Farazi and S. Behnke, "Online visual robot tracking and identification using deep LSTM networks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 6118–6125.
[40] X. Wang, G. Hua, and T. X. Han, "Discriminative tracking by metric learning," in *Proc. Eur. Conf. Comput. Vis.*, Crete, Greece, Sep. 2010, pp. 200–214.
[41] 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.*, Jun. 2014, pp. 1234–1241.
[42] W. Li, R. Zhao, T. Xiao, and X. Wang, "Deepreid: Deep filter pairing neural network for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 152–159.
[43] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556
[44] D. Cheng, Y. Gong, S. Zhou, J. Wang, and N. Zheng, "Person re-identification by multi-channel parts-based CNN with improved triplet loss function," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1335–1344.
[45] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 815–823.
[46] J. Dequaire, P. Ondrúška, D. Rao, D. Wang, and I. Posner, "Deep tracking in the wild: End-to-end tracking using recurrent neural networks," *Int. J. Robot. Res.*, vol. 37, nos. 4–5, pp. 492–512, 2017.
[47] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.
[48] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, 1960.
[49] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statist. Comput.*, vol. 10, no. 3, pp. 197–208, Jul. 2000.
[50] P. Ondruska and I. Posner, "Deep tracking: Seeing beyond seeing using recurrent neural networks," in *Proc. 13th AAAI Conf. Artif. Intell.*, Mar. 2016, pp. 3361–3367.
[51] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA, Neural Netw. Mach. Learn.*, vol. 4, no. 2, pp. 26–31, 2012.
[52] H. Kieritz, S. Becker, W. Hübner, and M. Arens, "Online multi-person tracking using integral channel features," in *Proc. 13th IEEE Int. Conf. Adv. Video Signal Based Surveill.*, Aug. 2016, pp. 122–130.
[53] L. Fagot-Bouquet, R. Audigier, Y. Dhome, and F. Lerasle, "Improving multi-frame data association with sparse representations for robust near-online multi-object tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 774–790.
[54] R. Sanchez-Matilla, F. Poiesi, and A. Cavallaro, "Online multi-target tracking with strong and weak detections," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 84–99.
[55] Y. Ban, S. Ba, X. Alameda-Pineda, and R. Horaud, "Tracking multiple persons based on a variational Bayesian model," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 52–67.
[56] W. Choi, "Near-online multi-target tracking with aggregated local flow descriptor," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3029–3037.
[57] 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, 2017.
[58] S. H. Rezatofighi, A. Milan, Z. Zhang, Q. Shi, and I. Reid, "Joint probabilistic data association revisited," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3047–3055.
[59] M. Yang and Y. Jia, "Temporal dynamic appearance modeling for online multi-person tracking," *Comput. Vis. Image Understand.*, vol. 31, pp. 16–28, Dec. 2016.
[60] Y. Xiang, A. Alahi, and S. Savarese, "Learning to track: Online multi-object tracking by decision making," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4705–4713.
[61] 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.*, Jun. 2016, pp. 1392–1400.
[62] 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.*, Mar. 2018, pp. 466–475.

JUN XIANG received the B.S. and M.S. degrees from the College of Electronics and Information Engineering, South-Central University for Nationalities, Wuhan, China, in 2008 and 2012, respectively, and the Ph.D. degree in pattern recognition and intelligent systems from the Huazhong University of Science and Technology, Wuhan, China. She is currently an Assistant Professor with the Department of Electronics and Information Engineering, South-Central University for Nationalities. Her research interests include deep learning, graphical models, image and object segmentation, and object tracking.

GUOSHUAI ZHANG received the B.S degree from the Department of Electronics and Information Engineering, Changchun University, in 2016. He is currently pursuing the M.S. degree with the Department of Electronics and Information Engineering, South-Central University for Nationalities, Wuhan, China.



JIANHUA HOU received the B.S. degree in optical engineering from the Beijing Institute of Technology, in 1985, the M.S. degree in optical engineering from the University of Electronic Science and Technology of China, in 1987, and the Ph.D. degree in pattern recognition and intelligent systems, Huazhong University of Science and Technology, Wuhan, in 2007. He is currently a Professor with the Department of Electronics and Information Engineering, South-Central University for Nationalities. His research interests include pattern recognition and computer vision.