An Online and Flexible Multi-Object Tracking Framework using Long Short-Term Memory

Xingyu Wan, Jinjun Wang, Sanping Zhou
Xi'an Jiaotong University
Institute of Artificial Intelligence and Robotics
28 West Xianning Road, Xi'an, Shaanxi, China, 710049

Abstract

The capacity to model temporal dependency by Recurrent Neural Networks (RNNs) makes it a plausible selection for the multi-object tracking (MOT) problem. Due to the non-linear transformations and the unique memory mechanism, Long Short-Term Memory (LSTM) can consider a window of history when learning discriminative features, which suggests that the LSTM is suitable for state estimation of target objects as they move around. This paper focuses on association based MOT, and we propose a novel Siamese LSTM Network to interpret both temporal and spatial components nonlinearly by learning the feature of trajectories, and outputs the similarity score of two trajectories for data association. In addition, we also introduce an online metric learning scheme to update the state estimation of each trajectory dynamically. Experimental evaluation on MOT16 benchmark shows that the proposed method achieves competitive performance compared with other state-of-the-art works.

1. Introduction

One of the key challenges at multi-object tracking (MOT) is to continuously and effectively model the vast variety of object appearances with high uncertainty in arbitrary scenarios, caused by occlusions, illumination variations, motion blur, false alarm and miss detections, variations of targets numbers, etc. Traditional methods to find the locations of target objects by using low-level handcrafted features [13, 41, 8, 12, 10] led to rather limited performance. Followed by the galloping progresses of deep learning techniques, many works have been proposed to utilize pre-trained models on a large-scale dataset to obtain richer feature representations based on convolutional neural networks (CNNs) [35] [11]. However, lack of sufficient training data and only rely on appearance feature leave the tracking issues unsolved. With the progress in object detection, "Tracking-by-detection" framework [23] has become a leading paradigm whereby the detection results of objects are represented as bounding boxes and available in a video sequence as prior information. Here, the tracking is casted as a problem of data association where the objective is to connect detection outputs into trajectories across video frames using reasonable measurements. Classical methodologies such as Multiple Hypothesis Tracking (MHT) [27] and the Joint Probabilistic Data Association Filter (JPDAF) [9] focused on establishing sophisticated models to capture the combinatorial complexity on a frame-by-frame basis. Later on, lots of works have been combining various of components such as motion dynamics and interaction information to obtain an effective state estimation [24, 32, 25, 7, 37, 29, 1].

Recently, Milan et al. [21] presented an end-to-end Recurrent Neural Network (RNN) for online multi-target tracking and confirmed that RNN-based approach can be utilized to learn complex motion models in realistic environments. In order to further explore the capability of combining both temporal and spatial components to model people trajectories using RNN, we introduce a Siamese LSTM network for metric learning, and thus introducing a new feature model for computing similarity between trajectories. By incorporating a window of the previous history of an object, LSTM is able to capture both linear and nonlinear features in a long-term, which should be more effective to analyze the objects' motion pattern. For each object, three cues of features are fused into one LSTM network for metric learning. Specifically, these cues are the people appearance learned by a re-id CNN, the motion represented as bounding boxes coordinates and the velocity of the object. Unlike the traditional metric learning using CNN, we argue that LSTM is capable of analyzing how these features evolve and keeping the sophisticated pattern as a memory for each object. After establishing the unique model for each trajectory, we introduce a Softmax layer for similarity computation between two objects. This enables our network to accomplish the tasks of metric learning and affinity computation jointly.

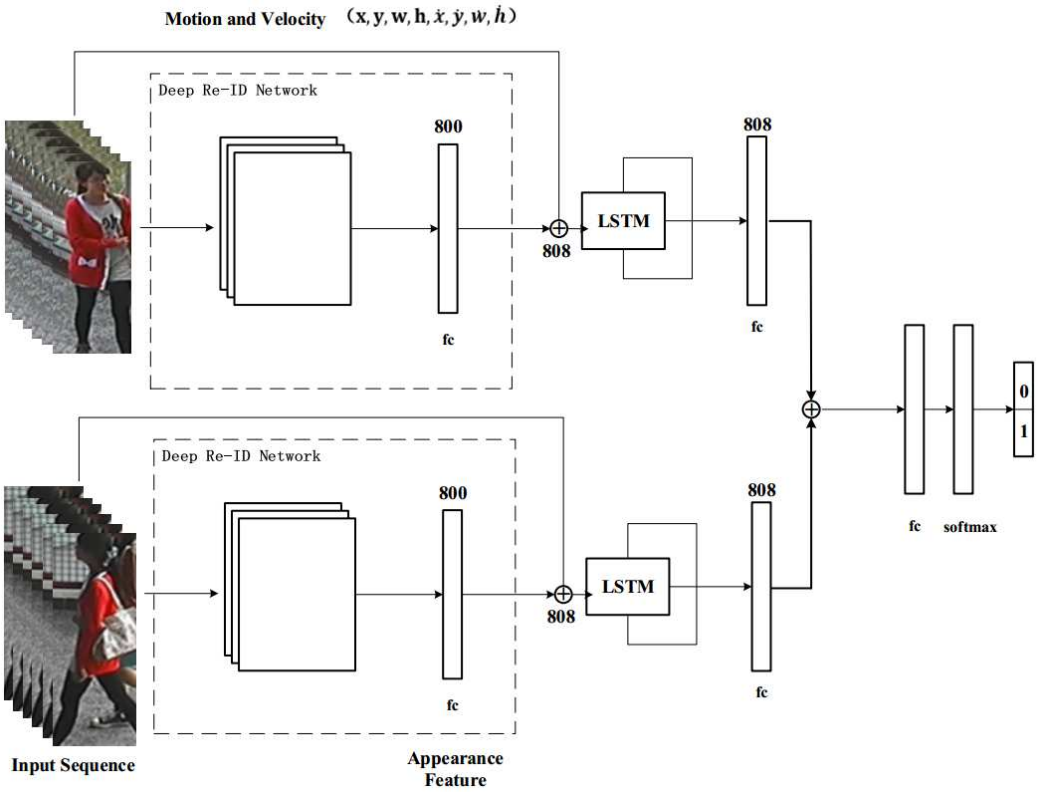

Figure 1. The architecture of our proposed MOT framework using LSTM

Our framework is showed as Fig.1.

Since offline training of an unified LSTM could not easily adapt to real scenario as each object has a separate motion pattern, we also introduce an online metric learning method to update the LSTM for each trajectory frame-by-frame. When assigning a trajectory to an existing one, we update the corresponding LSTM model by incorporating new features. In this way, feature representations of trajectories can be more accurate and update-to-date. We use two traditional yet robust methodologies as our initialization methods to obtain short tracklets for LSTM inputs, specifically the Kalman filter [14] along with Hungarian method [16], and the LK Optical Flow algorithm [20] along with IOU (intersection-over-union) distance computation. Our main contributions are as follows:

1) We fuse three features into one metric learning LSTM network for learning both temporal and spatial features, and at the same time our network can also output the similarity score for data association;

2) We propose an efficient and simple method for generating robust tracklets using Optical Flow and Affine Transformation, apart from the well-known Kalman Filter;

3) We introduce an online method to establish a discriminative trajectory model of each object;

4) Our proposed approach achieves a competitive tracking accuracy compared with other state-of-the-art methods, and our framework has the flexibility of any baseline methodologies which is applicable to arbitrary scenarios.

2. RELATED WORK

Within the ”Tracking-by-detection” paradigm, traditional data association techniques including the Multiple Hypothesis Tracker (MHT) [27] and the Joint Probabilistic Data Association Filter (JPDAF) [9] to solve the MOT problem is establishing sophisticated models to capture the combinatorial complexity on a frame-by-frame basis. Both methods have been revised recently in conjunction with a novel appearance model [15] or an efficient approximation [28] and shown great improvements in performance. Recently, a large amount of works have focused on casting the tracking problem as global optimization with simplified models. Flow network formulations [42] [26] [3] and probabilistic graphical models [39] [38] [2] [22] are often considered in this fashion, along with shortest-path, min-cost algorithms or even graph multicut formulations [33]. However, these methods are not applicable to online scenarios without seeing the future objects.

Learning an effective model for feature representation with corresponding similarity computation plays the central role in data association. Over the past decades, adopting

CNN-based feature representations for people appearance along with computation of the affinity between two measurements has been a popular trend. Nevertheless, merely rely on appearance feature can be problematic due to a simple fact that people with similar appearances are not necessarily identical. To this end, various trackers were taken to model different features of objects in the scene by incorporating a myriad of components such as motion, appearance, scale, etc. [24] [32] [25] [7] [37] [29] [1] have been combining the motion dynamics and interaction between measurements along with the target appearance. Only reason on adjacent temporal frames and combine these components linearly leads to a bottleneck of the feature representation methods.

Inspired by the success of Recurrent Neural Networks (RNNs) and their application to language modeling [34], some researchers have been trying to learn an end-to-end representation for state estimation utilizing RNNs [30] [21]. Sadeghian et al. [30] proposed an offline metric learning framework using a hierarchical RNN to encode long-term temporal dependencies across multiple cues, i.e., appearance, motion, and interaction. Milan et al. [21] presented an online RNN-based approach for multiple people tracking which is capable of performing prediction, data association and state update within a unified network structure. These works confirmed that RNN-based approach can be utilized to learn complex models in realistic environments, furthermore, LSTM networks are capable of tackling the one-to-one assignment task. Our work extend the research of RNN-based methods and leverage the power of LSTM for learning a discriminative model of object trajectory by integrating dynamic features both in temporal and spatial. Unlike training an unified model within a hierarchical framework in an offline fashion as [30], we introduce an online updating mechanism to establish dynamic models for each trajectory and achieve a state-of-the-art tracking performance.

3. Methodology

The proposed approach casts the MOT problem into four steps: feature learning, tracklets initialization, data association and trajectory updating, as presented in the following sections:

3.1. Feature Learning

Our overall architecture (Fig.1) includes both the metric learning component and the similarity computation component. For metric learning, the framework is consisted of two identical networks with a deep CNN and one LSTM layer. Three different features are integrated into one network for metric learning. We first employ a deep CNN pre-trained on a person re-identification dataset CUHK03 [19] proposed by Sanping et al. [43] to extract the 800 dimensional appearance feature. Once we generated the feature map of each object, we add the motion features represented by 4 bounding boxes coordinates as well as their corresponding velocities using a fully connected layer, then we obtain a fusion feature represented by a 808 dimensional vector which consist of 800 dimensional appearance feature, 4 dimensional motion feature and 4 dimensional velocity feature.

After this, we employ a LSTM layer for incorporating temporal dependencies of the tracking module. Different from RNNs, LSTM uses a memory cell containing a self-connected linear unit to prevent error signals from decaying quickly as they flow back in time. At any point in time, errors in the network (whether from cells or from gates) are used to drive weight changes. However, only the so-called constant error carousels (CECs) keep track of error as it flows back in time; errors elsewhere are truncated (errors outside CECs vanish exponentially fast anyway, just like in traditional RNNs). By tracking long-time scale dependencies in the CECs, LSTM is able to bridge huge time lags (1000 discrete time steps and more) between relevant events, while traditional RNNs already fail to learn in the presence of 10 step time lags, even with complex update algorithms such as real-time recurrent learning (RTRL) or back propagation through time (BPTT). In general, however, it is not sufficient to simply add linear counters to a RNN. Without some method of protecting the contents of these counters, such a network could quickly diverge. For this reason, LSTM CECs are arranged in memory blocks of cells that control the flow of information through the CECs. Though LSTM could in principle work with any differentiable protective mechanism, existing implementations use a small set of multiplicative gates as shown in Fig.2: an input gate learns to protect the CECs from irrelevant inputs, an output gate learns to turn off a cell block that is generating irrelevant output, and a forget gate allows CECs to reset themselves to zero when necessary. The mathematical expressions of this gated mechanism are shown in Eq.(1), Eq.(2), Eq.(3) and Eq.(4). Here i, o, f are the expressions of input, output and forget gates separately, c_t^l denotes the cell state of layer l at time t, h_t^l denotes the lth hidden layer at time t and $\bigodot$ represents element-wise multiplication.

$$i, o, f = \sigma(W^l(h_t^{l-1}, h_{t-1}^l)^\top) \tag{1}$$

$$g = \tanh[W^l(h_t^{l-1}, h_{t-1}^l)^\top] \tag{2}$$

$$c_t^l = f \bigodot c_{t-1}^l + i \bigodot g \tag{3}$$

$$h_t^l = o \bigodot \tanh(c_t^l) \tag{4}$$

For similarity computation, we have two streams of feature vectors flow into one softmax layer, so as to identify whether the two belong to a same identity or not. We use the Mean Squared Error (MSE) for model training as depicted by Eq.(5). Here n is the number of training samples

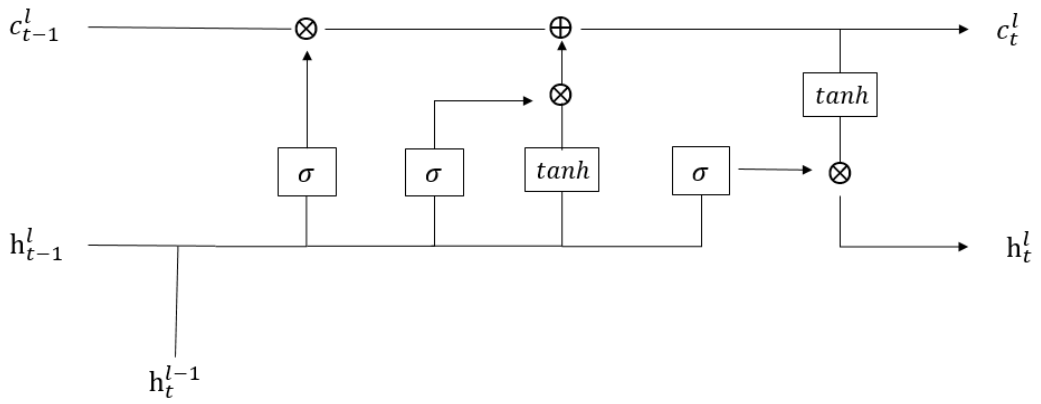

Figure 2. A LSTM memory block with one cell

in a batch, y_{label} is the ground truth value where 1 means two inputs are the same identity and 0 otherwise, y_{pred} is the output value of softmax and $\|\cdot\|$ is the squared Euclidean norm. We use the Adam method for stochastic optimization.

$$L_{MSE} = \frac{1}{n}\sum_{i=1}^{n} \|y_{label} - y_{pred}\|_2^2 \tag{5}$$

3.2. Tracklets Initialization

For the sake of learning long-term features using LSTM and computing the affinity of two trajectories, we need to initialize short tracklets as the inputs of our LSTM network. We employ two separate algorithms for initialization, specifically the Kalman filter [14] along with Hungarian method [16], and the LK Optical Flow algorithm [20] along with IOU distance computation.

As described in [5], using Kalman filter for prediction and updating, along with Hungarian method for assigning multiple labels, facilitates both efficiency and reliability for online tracking. We establish the motion model as Eq.(6), here x,y are the bbox coordinates of center point, and w, h are the weight and height of the bbox. The state estimation and data association approaches are the same as [5]. More specifically, we predict the object location in next frame via a Kalman filter by solving the velocity components optimally, then we compute the assignment cost matrix with the intersection-over-union (IOU) distance and solve the assignment optimally using the Hungarian algorithm. When a detection is associated to a target, we update the target state using the detected bounding box. If no detection is associated to the target, its state is simply predicted without correction using the linear velocity model. Here we set a rather high threshold value IOU_{min} in order to obtain short but reliable tracklets.

$$X = [x, y, w, h, \dot{x}, \dot{y}, \dot{w}, \dot{h}]^T \tag{6}$$

Since using Kalman Filter for state determination is already a mature methodology for visual tracking, which provides a recursive solution to the linear optimal filtering problem based on the established motion model, here we introduce a simple and efficient initialization approach to distinguish the feature learning process of motion from Kalman Filter. Optical Flow presents an apparent change of a moving object's location or deformation between frames, further more, Optical Flow estimation yields a two-dimensional vector filed, i.e., motion field, that represents velocities and directions of each point of an image sequence. We argue that using Optical Flow along with Hungarian algorithm is a simple and feasible approach for visual tracking and can hence generate robust tracklets. The overall visual tracking approach using Optical Flow is illustrated in Fig.3. Giving the previous and the current image frames I^{t-1}, I^t, sparse local optical flow information $V^{t-1,t}(x, y)$ can be derived. Specifically, the optical flow determination is solved by the calculation of partial derivatives of the image signal using Lucas-Kanade method [20]. And the propagated position $\hat{d}_i^t$ of point i in the frame t is

$$\hat{d}_i^t = V^{t-1,t}(d_i^{t-1}) = d_i^{t-1} + v_i^{t-1,t} \tag{7}$$

where $v_i^{t-1,t}$ is the local displacement for d_i^{t-1}. We then compute the affine transformation of inner points of bbox between two adjacent frames:

$$f(\hat{d}_i^t(x', y'), p_i^{t-1}(x, y)) = \begin{cases} x' = \alpha_1 x + \beta_1 y + \gamma_1 \\ y' = \alpha_2 x + \beta_2 y + \gamma_2 \end{cases} \tag{8}$$

Here $\hat{d}_i^t(x', y')$ is 2D propogated position of point i at frame t, and $p_i^{t-1}(x, y)$ is the same point i at frame $t-1$, α, β and γ are the affine transformation coefficients. The predicted location $B\hat{B}ox^t$ is obtained by fitting the previous bbox coordinates into the affine transformation: $f(BBox^{t-1}, B\hat{B}ox^t)$. Other than this, we employ the same assignment strategy as above which is computing cost matrix using IOU distance and make the assignment using

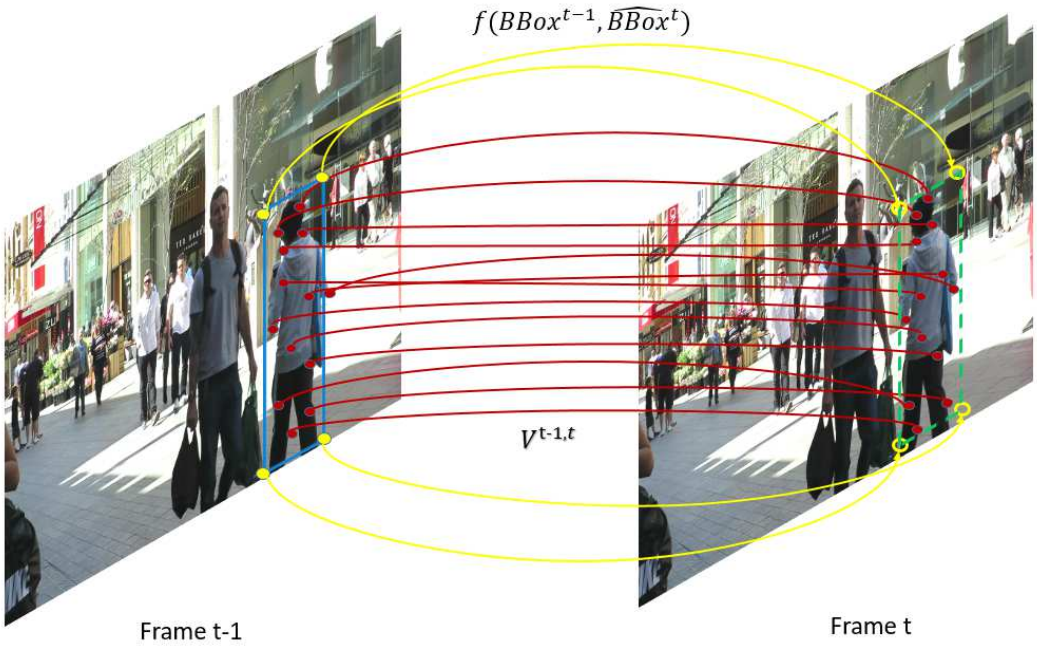

Figure 3. The illustration of visual tracking approach using Optical Flow. The red lines are optical flow information, yellow lines are affine transformation and the green dashed lines at frame t is predicted location.

Hungarian algorithm. More specially, we assume that detections of an object in consecutive frames hava an unmistakably high overlap IOU (intersection-over-union) which is commonly the case when using sufficiently high frame rates. The IOU measure used in our approach is defined as:

$$IOU(a,b) = \frac{Area(a) \bigcap Area(b)}{Area(a) \bigcup Area(b)} \tag{9}$$

We first compute the IOU distance $IOU(B\hat{B}ox_i^t, D_i^t)$ between the predicted location $B\hat{B}ox_i^t$ of target i with its neighborhood detections $\{d_k^t\} \in D_i^t$ at frame t. We pick the detection result with the max IOU value higher than IOU_{min} as the candidate, if all the $IOU(B\hat{B}ox_i^t, D_i^t)$ are lower than IOU_{min}, we just use the predicted location $B\hat{B}ox_i^t$ as the candidate. We then compute the IOU distance between the detection results at frame $t-1$ and the candidate locations at frame t for each target. Followed by this, the assignment problem leads to an optimal association between detections and candidates which can be solved by applying the Hungarian algorithm maximizing the sum of all IOUs at frame t. All detections not assigned to an existing track will start a new one. All tracks without an assigned detection will end.

3.3. Data Association and Trajectory Updating

As described above, our network is able to output the similarity score of two trajectories. Therefore, after we obtain numbers of short yet reliable tracklets in time order with initialization, we input these tracklets into our pre-trained LSTM network for affinity computation. To be specific, for each tracklet $A^i\{t+1, t+2, ...t+l\}$, we put a window of frames with length l from the end of tracklet into one stream of our Siamese network as an anchor. For those tracklets whose first frames are within $\{t+l+1, t+l+2, ..., t+l+\alpha\}$, we take the same length l of frames from the beginning of tracklets and input them into another stream of our network as candidates. Here the parameter α indicates the time gap, if the interval of two tracklets are longer than α, we take these two targets not related. Our siamese LSTM network pre-trained on the training dataset is then taken into a forward propagation to obtain the similarity scores of the anchor and all its candidates. The output of LSTM network is a single value between 0 and 1 which indicates the affinity of two tracklets. We set a threshold value S_{min} to filter out the candidates with low confidence. If the output is lower than S_{min}, we take the target as different identity with the anchor and remove it from the candidates. For those candidates with similarity scores higher than S_{min}, if there exist time overlaps, we just use the tracklets ahead in time and remove others from the candidates. After we compute the similarity scores of all anchors with their corresponding candidates at one time step, we employ the Hungarian algorithm to solve the global optimal problem for our data association and thus obtain the longer tracklets. For those candidates assigned to the anchors, we mark them as matched pairs, and for all those not assigned ones, we take them as unmatched pairs.

Table 1. Evaluation results of the proposed method on MOT16 benchmark.

Method	MOTA↑	MOTP↑	MT↑	ML↓	FP↓	FN↓	IDS↓	Frag↓
EAMTT	52.5	78.8	19.0%	34.9%	**4407**	81223	910	1321
SORT	59.8	**79.6**	25.4%	22.7%	8698	63245	1423	1835
Deep SORT	61.4	79.1	**32.8**%	**18.2**%	12852	56668	**781**	2008
IOU	57.1	77.1	23.6%	32.9%	5702	70278	2167	3028
LSTM+Kalman filter	**62.6**	78.3	32.7%	21.1%	10604	**56182**	1389	1534
LSTM+Optical Flow	62.1	79.1	32.5%	29.1%	6179	62206	793	**832**

Using an unified network trained offline could not be easily adapted to real scenario as each object has a separate motion pattern. Thus we introduce an online updating approach to retrain the LSTM for each tracklet frame-by-frame. That is to say, when assigning a tracklet to an existing one, we train the LSTM model again to update the metric by adding new feature vectors of appearance, motion and velocity. More specifically, we initialize new LSTM models for all targets at first, and when the process of data association at one time step is accomplished, we pick up training samples from matched pairs with label 1 along with unmatched pairs with label 0. Then we input these training samples into the Siamese networks to train one more time and thus obtain the updated feature representation for each target. In general, each trajectory is initialized with a pre-trained model and each model will be updated when an assignment occurs. The data association and model updating is proceeded interactively within a time step. In this way, feature representations of trajectories can be more accurate and update-to-date.

4. Experiments

4.1. Implementation Details

We implemented our framework in Python using Tensorflow, with six cores of 2.4GHz Inter Core E5-2680 and three NVIDIA GTX 1080 GPUs. The input of our network for metric learning is a sequence of cropped images for each pair of trajectories which are resized to 224×224. Here we set the window length l which is also called timestep of LSTM input to 6 and the number of LSTM hidden layers is 20. At the stage of initialization, we set an uniform threshold value IOU_{min} to 0.8 for both approaches. At the stage of data association, we set the parameter S_{min} to 0.7 and parameter α to 30. Moreover, to compare with other state-of-the-art online tracking algorithms like SORT [5], we employ a private detector provided by Yu et al. [40].

4.2. Benchmark Results

Our proposed method was evaluated with a set of testing sequences from the MOT challenge [17] database which contains both moving and static camera sequences. We evaluated our tracking performance using the standard MOT metrics [4]: Multi-Object Tracking Accuracy (MOTA↑), Multi-Object Tracking Precision (MOTP↑), Mostly Tracked targets (MT↑), Mostly Lost targets (ML↓), False Positives (FP↓), False Negatives (FN↓), ID Switches (IDS↓) and Fragments (Frag↓). Evaluation measures with (↑) means that higher scores denote better performance, while evaluation measures with (↓) means that lower scores denote better performance.

Table 1 show the quantitative results of our method on MOT16 benchmark. For a fair comparison, we only list the most relevant online trackers with state-of-the-art accuracy. The baseline method EAMTT proposed by [31] is an online multi-target tracker which exploits strong and weak detections in a Probability Hypothesis Density Particle framework. As discussed above, our proposed method use two different initialization approaches. Specifically, as shown in table 1, using Kalman filter [14] along with Hungarian algorithm [16] as initialization for our method is called "LSTM+Kalman filter", while using LK Optical Flow algorithm [20] along with IOU distance computation as initialization is called "LSTM+Optical Flow". Generally, both two proposed methods achieved the highest MOTA scores compared to other online trackers. As described in [18], MOTA is the measure that best aligns with the human visual assessment (HVA), and Mostly Tracked (MT) follows as second-best measure. Higher than other trackers in MOTA, our method also gains a competitive MT score of 32.7% which is very close to the highest one (32.8%). In comparison to SORT [5] whose core framework is using Kalman filter [14] for tracking and Hungarian algorithm [16] for assignment, MOTA score of our method "LSTM+Kalman filter" increases from 59.8 to 62.6. And compared with the tracker IOU [6] which also uses the strategy of introducing Optical Flow algorithm with IOU distance computation, the MOTA score of our method "LSTM+Optical Flow" increases from 57.1 to 62.1. That is to say, due to the online metric learning using LSTM, we successfully improved the tracking performance of baseline methods. And we notice that the ID Switches (IDS) of both our methods (1389 and 793) do not gain a significant decline compared to other state-of-the-art trackers (781). That is because instead of focusing on filling the gaps to obtain long trajectories, our trackers take more interest in exploiting feasible information to better recover trajectories. Specifically, our trackers tend to update the feature representation of targets dynamically when they are visible in the scene, and initialize new representation when they reappear after a long occlusion which

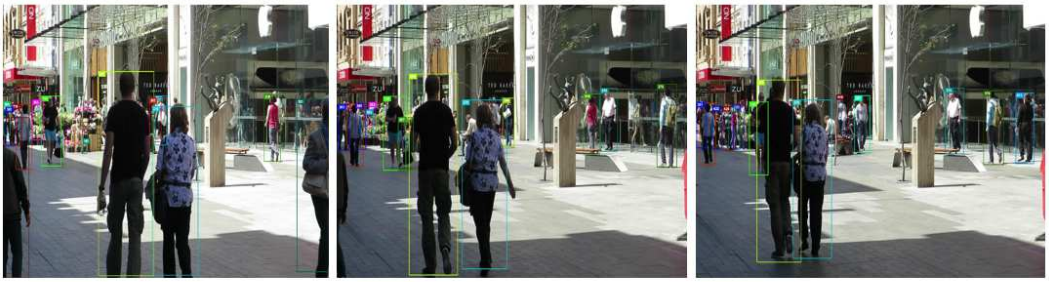

Figure 4. The tracking results of our proposed method on selected MOTChallenge sequence. Our trackers are able to maintain identities through a rather long variation.

is highly related to the parameter α. This can also be verified from the quantitative results that we achieve the lowest Fragments (Frag) among all the state-of-the-art trackers, which means by fusing both temporal and spatial features with LSTM and update the models for each trajectory, our tracker is able to maintain identities through longer and more intense variations. An exemplary illustration of our tracking performance is shown in Fig.4.

Due to the rapid development of sensor technology, nonvisible range sensors have raised lots of researchers' attention in both academia and industry. Many investigators have suggested that use of TIR (thermal infrared) images which typically operate in the midwave (3 to 5μm) or long-wave (8 to 14μm) can provide information on important cultural, geological, and agricultural variables. As for our work, [36] has published a thermal infrared video benchmark called TIV for various visual analysis tasks which include multi-object tracking. Lack of RGB appearance feature, our proposed algorithm can not be adapted to infrared videos directly. As mentioned before, our proposed framework fuses three cues into one network for feature learning. We now intend to replace the RGB feature by thermal infrared one, and pay more attention to the motion component in feature learning phase. More experimental evaluations and analysis on infrared videos especially the TIV benchmark will be demonstrated in the future.

5. Conclusions

MOT using RNNs has drawn more and more attention in recent years. As an extension of researching RNN-based methods, we present a novel Siamese LSTM network for metric learning along with an online updating scheme for data association based MOT, by leveraging the power of LSTM. Our proposed network fuses three most relevant features of trajectories into one LSTM to interpret both temporal and spatial components nonlinearly, and is able to output the similarity score at the same time. Furthermore, we initialize a LSTM model for each trajectory and update the metric in an online fashion during the tracking phase. And we also introduce an efficient and feasible visual tracking approach using Optical Flow and affine transformation, which can generate robust tracklets for our initialization. The presented MOT framework achieves state-of-the-art tracking accuracy, and as shown in the experiments, the improvement on performance confirms that our method has the flexibility to be applied to arbitrary scenarios.

References

[1] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–971, 2016.

[2] A. Andriyenko, K. Schindler, and S. Roth. Discrete-continuous optimization for multi-target tracking. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1926–1933. IEEE, 2012.

[3] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *IEEE transactions on pattern analysis and machine intelligence*, 33(9):1806–1819, 2011.

[4] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *Journal on Image and Video Processing*, 2008:1, 2008.

[5] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft. Simple online and realtime tracking. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 3464–3468. IEEE, 2016.

[6] E. Bochinski, V. Eiselein, and T. Sikora. High-speed tracking-by-detection without using image information. In *Advanced Video and Signal Based Surveillance (AVSS), 2017 14th IEEE International Conference on*, pages 1–6. IEEE, 2017.

[7] W. Choi and S. Savarese. Multiple target tracking in world coordinate with single, minimally calibrated camera. In *European Conference on Computer Vision*, pages 553–567. Springer, 2010.

[8] M. Danelljan, G. Häger, F. Khan, and M. Felsberg. Accurate scale estimation for robust visual tracking. In *British Machine Vision Conference, Nottingham, September 1-5, 2014*. BMVA Press, 2014.

[9] T. Fortmann, Y. Bar-Shalom, and M. Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE journal of Oceanic Engineering*, 8(3):173–184, 1983.

[10] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):583–596, 2015.

[11] S. Hong, T. You, S. Kwak, and B. Han. Online tracking by learning discriminative saliency map with convolutional neural network. In *International Conference on Machine Learning*, pages 597–606, 2015.

[12] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao. Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 749–758, 2015.

[13] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *IEEE transactions on pattern analysis and machine intelligence*, 34(7):1409–1422, 2012.

[14] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[15] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg. Multiple hypothesis tracking revisited. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4696–4704, 2015.

[16] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics (NRL)*, 2(1-2):83–97, 1955.

[17] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler. Motchallenge 2015: Towards a benchmark for multi-target tracking. *arXiv preprint arXiv:1504.01942*, 2015.

[18] L. Leal-Taixé, A. Milan, K. Schindler, D. Cremers, I. Reid, and S. Roth. Tracking the trackers: an analysis of the state of the art in multiple object tracking. *arXiv preprint arXiv:1704.02781*, 2017.

[19] W. Li, R. Zhao, T. Xiao, and X. Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 152–159, 2014.

[20] B. D. Lucas, T. Kanade, et al. An iterative image registration technique with an application to stereo vision. 1981.

[21] A. Milan, S. H. Rezatofighi, A. R. Dick, I. D. Reid, and K. Schindler. Online multi-target tracking using recurrent neural networks. In *AAAI*, pages 4225–4232, 2017.

[22] A. Milan, K. Schindler, and S. Roth. Detection-and trajectory-level exclusion in multiple object tracking. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3682–3689. IEEE, 2013.

[23] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 4293–4302. IEEE, 2016.

[24] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 261–268. IEEE, 2009.

[25] S. Pellegrini, A. Ess, and L. Van Gool. Improving data association by joint modeling of pedestrian trajectories and groupings. In *European conference on computer vision*, pages 452–465. Springer, 2010.

[26] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1201–1208. IEEE, 2011.

[27] D. Reid. An algorithm for tracking multiple targets. *IEEE transactions on Automatic Control*, 24(6):843–854, 1979.

[28] S. H. Rezatofighi, A. Milan, Z. Zhang, Q. Shi, A. R. Dick, and I. D. Reid. Joint probabilistic data association revisited. In *ICCV*, pages 3047–3055, 2015.

[29] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *European conference on computer vision*, pages 549–565. Springer, 2016.

[30] A. Sadeghian, A. Alahi, and S. Savarese. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. *arXiv preprint arXiv:1701.01909*, 4(5):6, 2017.

[31] R. Sanchez-Matilla, F. Poiesi, and A. Cavallaro. Online multi-target tracking with strong and weak detections. In *European Conference on Computer Vision*, pages 84–99. Springer, 2016.

[32] P. Scovanner and M. F. Tappen. Learning pedestrian dynamics from the real world. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 381–388. IEEE, 2009.

[33] S. Tang, M. Andriluka, B. Andres, and B. Schiele. Multiple people tracking by lifted multicut and person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3539–3548, 2017.

[34] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 3156–3164. IEEE, 2015.

[35] N. Wang, S. Li, A. Gupta, and D.-Y. Yeung. Transferring rich feature hierarchies for robust visual tracking. *arXiv preprint arXiv:1501.04587*, 2015.

[36] Z. Wu, N. Fuller, D. Theriault, and M. Betke. A thermal infrared video benchmark for visual analysis. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, pages 201–208. IEEE, 2014.

[37] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg. Who are you with and where are you going? In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1345–1352. IEEE, 2011.

[38] B. Yang and R. Nevatia. Multi-target tracking by online learning of non-linear motion patterns and robust appearance models. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1918–1925. IEEE, 2012.

[39] B. Yang and R. Nevatia. An online learned crf model for multi-target tracking. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2034–2041. IEEE, 2012.

[40] F. Yu, W. Li, Q. Li, Y. Liu, X. Shi, and J. Yan. Poi: multiple object tracking with high performance detection and appearance feature. In *European Conference on Computer Vision*, pages 36–42. Springer, 2016.

[41] J. Zhang, S. Ma, and S. Sclaroff. Meem: robust tracking via multiple experts using entropy minimization. In *European Conference on Computer Vision*, pages 188–203. Springer, 2014.

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

[43] S. Zhou, J. Wang, D. Meng, X. Xin, Y. Li, Y. Gong, and N. Zheng. Deep self-paced learning for person re-identification. *Pattern Recognition*, 76:739–751, 2018.