Multi-Object Tracking with Multiple Cues and Switcher-Aware Classification

Weitao Feng[1] Zhihao Hu[1,2] Wei Wu[1] Junjie Yan[1] Wanli Ouyang[3]

[1]Sensetime Group Limited
[2]Beihang University
[3]The University of Sydney
{fengweitao, huzhihao, wuwei, yanjunjie}@sensetime.com
wanli.ouyang@sydney.edu.au

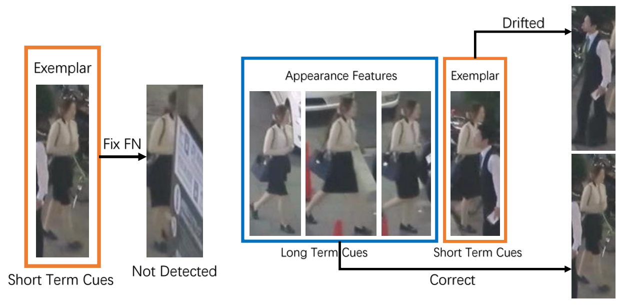
Abstract

In this paper, we propose a unified Multi-Object Tracking (MOT) framework learning to make full use of long term and short term cues for handling complex cases in MOT scenes. Besides, for better association, we propose switcher-aware classification (SAC), which takes the potential identity-switch causer (switcher) into consideration. Specifically, the proposed framework includes a Single Object Tracking (SOT) sub-net to capture short term cues, a re-identification (ReID) sub-net to extract long term cues and a switcher-aware classifier to make matching decisions using extracted features from the main target and the switcher. Short term cues help to find false negatives, while long term cues avoid critical mistakes when occlusion happens, and the SAC learns to combine multiple cues in an effective way and improves robustness. The method is evaluated on the challenging MOT benchmarks and achieves the state-of-the-art results.

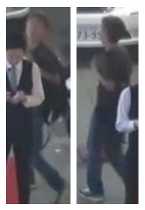
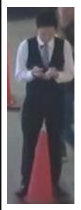
1. Introduction

Multi-Object-Tracking (MOT) is important in video analysis systems, such as video survelliance and self-driving car. It aims to maintain trajectories of all targets from categories of interest. The most recent methods in MOT follow the tracking-by-detection paradigm, which takes the frame-wise detections as the input and associates detections as the final trajectories. However, the detections are not always accurate enough, which could substantially influence the tracking. Besides, the occlusion and abnormal motion are another two problems in MOT.

We define two different cues in MOT. The short term cues mean updated cues between neighbouring frames, which include current target position, appearance and mo-



(a) (b)

Pair to consider	Switcher	Matching Confidence	
		With Switcher	Without Switcher
		0.7 High Confidence	0.2 Low Confidence

(c)

Figure 1. (a) False negative (FN): detector may not detect the occluded target, while SOT tracker can find this target to complement the detector. (b) Occlusion: when occlusion happens, the updated appearance is similar to the wrong target, and SOT is easy to drift. In contrast, the overall appearance of the tracklet is still stable and more reliable for association. (c) Switcher helps matching: without switcher, the matching confidence for the correct pair is low, while the matching score is higher when the classifier is aware of the occlusion situation and appearance information of the switcher.

tion. The long term cues stand for tracklet-long cues containing appearance features of the object within the tracklet. The recent single object tracking (SOT) approaches with high performance can be used in MOT for capturing short term cues, which are helpful when handling inaccurate detection results and abnormal motion. As shown in Figure 1(a), SOT trackers are effective to reduce false negatives (FN). Though short-term cues are helpful in many cases, most short term cues become unreliable when occlusions happen because the inclusion of occluded region makes the SOT tracker drift. Then long term cues of tracklet appearance can help to avoid the drift in SOT caused by occlusion. For the example shown in Figure 1(b), long term cues are still stable when occlusion happens.

Previous works did not make the fully use of the two cues. Many works like [42] that include SOT tracker in MOT did not consider combining SOT results in data association, while some works like [30] did not use SOT trackers to handle short term cues. Other rule based combination of long term and short term cues like [39] cannot learn from data with different situations and may over-fit to some specific cases. It also is a question for learning effective combination of short term and long term cues. In previous experiments (see Sec. 4.3), we have observed that it is hard to combine all cues in one network. That is, SOT tracker for short term cues cannot distinguish similar objects, and the network for long term cues cannot predict the precise position of target. Based on this motivation, we propose a unified MOT framework to generate short term and long term cues, as well as adaptively choose them for data association.

Another motivation of this paper is to use local interaction information to solve identity-switches. We have found that the potential identity-switch causer (switcher) is critical for correct matching. For example, Figure 1(c) shows how the switcher helps matching. Driven by this motivation, we use a switcher-aware classifier, which is implemented using boosting decision trees, to encode potential switcher information and improve the tracking robustness.

This paper proposes MOT approach with multiple cues and switcher-aware classification. The state-of-the-art method of SOT is used for capturing short term cues and a re-identification (ReID) method is applied for extracting long term cues. During data association, the switcher-aware classifier gathers all long term and short term cues and takes potential switcher into consideration, then generates scores building a bipartite graph for matching.

The main contributions of our work are listed as follows:

1. An effective MOT framework learning to capture long term and short term cues and making adaptive decisions for robust tracking.

2. A switcher-aware classification (SAC) in data association for improving the robustness of MOT to identity switch. We also introduce a simple but effective approach to search for potential switcher.

Extensive experimental results on both MOT16 and MOT17 benchmarks[26] clearly show the effectiveness of the proposed framework.

2. Related Work

2.1. MOT Using SOT Tracker

Some previous works [42, 10, 37, 38] have tried to apply SOT trackers into MOT task. Chu *et al.* [10] uses CNN-based single object tracker and handles drift through a spatial-temporal attention mechanism, it regards all detections as SOT proposals. Xiang *et al.* [37] utilizes MDP method to track targets in tracked state with optical flow. Most works have never benefited from the progress of visual object tracking (VOT) task. In recent years significant progress has been made in the single object tracking field. Trackers like GOTURN[14], Siamese-FC[4], ECO[11], Siamese-RPN[22] have highly improve the tracking accuracy. Method proposed by [42] directly applies the ECO-HC[11] tracker from visual object tracking task with a cost-sensitive loss and designed a spatial-temporal network for data association when SOT tracker is considered losing the target. However, an online-updating SOT tracker is slow in speed and costs a lot of memory. While offline training siamese SOT trackers like Siamese-RPN[22] reach the state-of-the-art accuracy at a high speed of more than 80 frames(targets) per second. More importantly, most methods have not combined cues generated from SOT tracker with other cues. They separate the SOT tracker and the data association process. Different from these works, we use the information from SOT together with long term cues from the trajectory for learning to associate detection/tracking results. Our usage of long term cues helps to solve the problems of drift in SOT, which cannot be solved effectively in existing SOT for MOT approaches.

2.2. Data Association

Data association is an important procedure of all tracking-by-detection-based MOT methods. [41, 33, 34, 35, 29] formulate the data association process as various optimization problems. Most of them are variants of graph segmentation problem and they need batch processing. Most online processing methods use Hungarian Algorithm[28] or minimum-cost-network-flow to solve a bipartite graph matching problem and they are effective.

Some works like [32, 30, 27, 16, 1, 2] emphasize to improve the features used in data association. [30, 27, 19] exploit RNNs in the MOT task. Sadeghian *et al.* [30] combines appearance, motion and interaction cues into a unified RNN network. Milan *et al.* [27] focuses on the utilization of positions and motions of the targets. Son *et al.* [32] develops a new training method with ranking loss and regression loss to obtain higher accuracy.

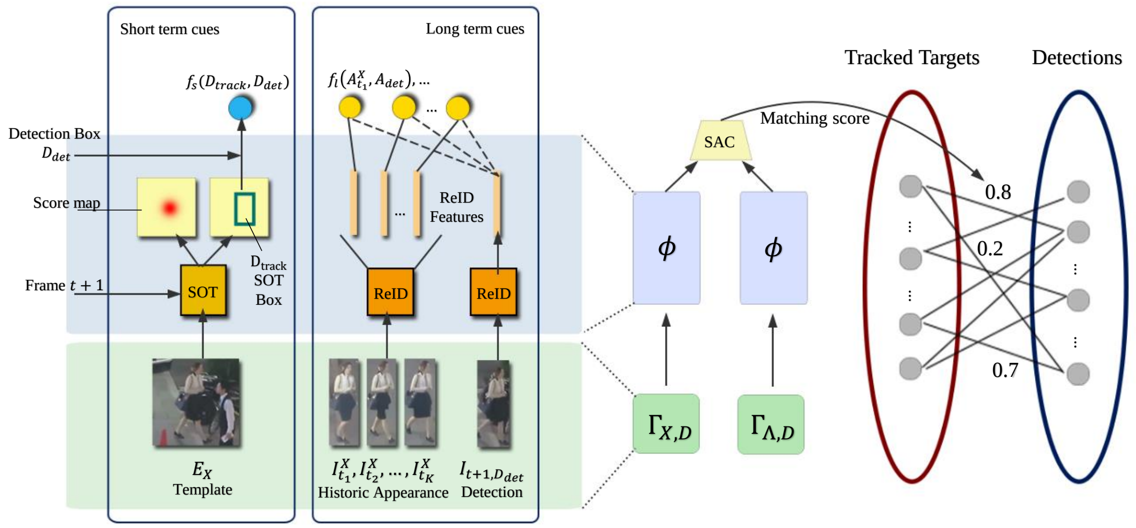


Figure 2. Overall design of the proposed MOT framework. Short term cues are captured by SOT sub-net and long term cues are captured by ReID sub-net. The switcher aware classifer (SAC) predicts whether the detection result matches the target using the short term and long term cues of the target, the detection result, and the switcher.

However, the disadvantages of these methods are well-known. First, they did not use SOT tracker to deal with inaccurate detection results, especially false negatives. Second, although they try a lot of discriminative features, they seldom take the local position and interaction information into training or inference phase. Sadeghian *et al*. [30] use spatial information from neighboring identities. But they do not use SOT tracker or the appearance information from switcher. Some works like [24, 39] ensemble appearance features with position and motion features. Their designs in using these information sources are based on heuristic rules, but not based on principle learning. Most of these works just regard data association as a pairwise matching problem between single tracklet and the detection. It is obvious that we will lose the valuable local interaction information which may indicate some critical errors. Though these mistakes will not cause huge drawback on target recall or precision, e.g., not a drawback in primary metric MOTA[3], somehow they influence the robustness of a tracking system and are significant to its application. In this work, we introduce the SAC for robust tracking which takes the most possible switcher into training and inference. And we combine the short term cues and long term cues in a balanced and data-driven way.

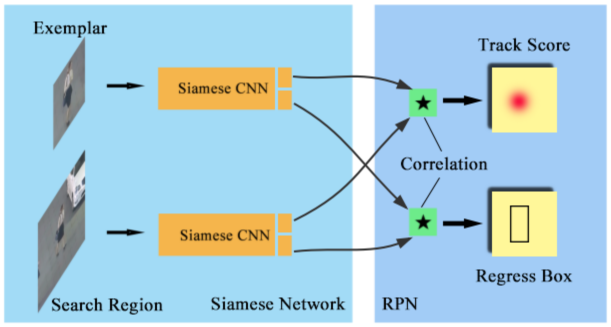
3. The Proposed Framework

Problem Formulation. The trajectory of one tracked target can be denoted by $X = \{X_t\}$, where $X_t = [X^x_t, X^y_t, X^w_t, X^h_t]$, t is the frame index, X^x_t, X^y_t is the top-left coordinate of the bounding box, and X^w_t, X^h_t is the width and height of the bounding box. q_X is the overall tracking quality for target X and I^X_t is the appearance of target X at frame t.

3.1. Overall Design

Figure 2 shows the overall design of the proposed MOT framework. The framework uses the following steps for online mode:

- Step 1. Initially, the set $\mathcal{S}$ of tracked targets is empty and $t = 1$.
- Step 2. At frame t, for a target X, the template E_X of the target is sought in the next frame I_{t+1} by using the SOT sub-net. The SOT sub-net outputs the most possible location D_{track} for the template in I_{t+1}.
- Step 3. For a detection result D_{det} in I_{t+1}, its corresponding detection image region $I_{t+1,D_{det}}$ and the historical appearances of the target $\{I^X_{t_i}\}, i = 1, 2, ..., K$ will be used by the ReID sub-net to extract the long-term ReID features.
- Step 4. The location of the target D_{track} found by SOT in Step 2, the location D_{det} found by the detector, the ReID features obtained in step 3 will be combined into the matching feature of the target.
- Step 5. Find the potential switcher of the target Λ, i.e., the most possible identity switch causer, and extract its SOT and ReID features.
- Step 6. With the aid of the matching features of the switcher, the matching features of the target are used by the switcher-aware classifier (SAC) to generate the matching score on whether the detection result should



(a)

Figure 3. Siamese-RPN architecture for SOT.

Type	Kernel/Stride	Output Size	Padding
input	- / -	$3 \times 224 \times 224$	-
convolution	7×7 / 2	$29 \times 112 \times 112$	3
pool	3×3 / 2	$29 \times 56 \times 56$	-
convolution	1×1 / 1	$27 \times 56 \times 56$	-
convolution	3×3 / 1	$142 \times 56 \times 56$	1
pool	3×3 / 2	$142 \times 28 \times 28$	-
inception-A	- / -	$379 \times 28 \times 28$	-
inception-A	- / -	$679 \times 28 \times 28$	-
pool	3×3 / 2	$679 \times 14 \times 14$	-
inception-A	- / -	$1037 \times 18 \times 18$	-
inception-A	- / -	$1002 \times 18 \times 18$	-
inception-A	- / -	$938 \times 18 \times 18$	-
inception-A	- / -	$861 \times 18 \times 18$	-
pool	14×14 / 1	$861 \times 1 \times 1$	-
fc	- / -	256	-

Table 1. The modified Inception-v4 architecture.

match the target or not. This step is repeated for each detection result in the frame I_{t+1} in order to obtain their matching scores to the tracked target.

- Step 7. Build a bipartite graph of tracked targets and the detection results using the matching scores between the tracked targets and the detection results found in Step 6. Find the matching results of the graph using minimum-cost-network-flow.
- Step 8. For the matched targets, update the position and template using the information of the matched detection. For targets not matched, update tracklet position using SOT results, and drop targets which are considered unreliable or lost. For isolated detection results, if their confidence scores satisfy the condition of new target, they will be added to the set of tracked targets.
- Step 9. Repeat steps from 2 to 8 for the next frame by setting $t = t + 1$, until no more frames arrive.

3.2. Using SOT Tracker for Short Term Cues

Baseline tracker. We use Siamese-RPN tracker [22] in our framework for extracting short term cues. Following the original schema, the template E_X at the current frame, also called the exemplar, is resized to 127×127. To search the target at the next frame, search region R is cropped from frame I_{t+1} according to the position of X. The search region is then resized to 255×255. Specifically, the picture scale of the search region is the same as the exemplar. As shown in Figure 3, the search region and the exemplar are passed through the shared-weight siamese CNN. The CNN features of the exemplar are then used by two branches, each consisting of two convolution layers, similarly for the search region. One branch for obtaining the score maps and the other for bounding box regression. The correct location of the exemplar in the search region is supposed to have largest score in the score map. The bounding box regression at different locations should point to this correct location.

Short term features generation. The SOT sub-net outputs a SOT score and the predicted bounding box, called SOT box. The detection bounding box to be matched is called detection box. We denote the SOT score as p, the SOT box as D_{track}, the detection box as D_{det}, then short term feature f_s is calculated as following:

$$f_s(D_{track}, D_{det}) = IoU(D_{track}, D_{det}) \tag{1}$$

Distractor-aware SOT tracker for MOT tracking. In order to maximize the effect of Siamese-RPN. We modify the anchors to fit the target scales of pedestrian. Besides, we refine the network using pedestrian data. Another problem of the SOT tracker is that it is hard to tell when to stop tracking if the target is lost. When the tracker drifts to background distractor, the tracker may not be able to stop tracking the distractor. To make the tracker score distractor-aware, we design a tracking score refine strategy. We refine the tracker score using the matching results found at the step 7 introduced in Sec. 3.1. For target X, the refined overall tracking quality q_X, is as following:

$$q_{X,t+1} = \begin{cases} \frac{q_{X,t} + IoU(D_{track}, D_{det}) \cdot p}{2}, & \text{if matched,} \\ q_{X,t} \cdot decay \cdot p^k, & \text{otherwise,} \end{cases} \tag{2}$$

where $decay$ and k are hyper-parameters used for dealing with inconsistent targets, D_{det} is the detection box. In this way, we can drop unreliable targets if the tracking quality q_X is below a threshold ζ_t.

3.3. Using ReID Network for Long Term Cues

We use a modified version of GoogLeNet Inception-v4 as the backbone CNN of the ReID sub-net. The ReID feature is extracted from the last FC layer before classification. Table 1 and Table 2 show details of the backbone CNN.

Quality-aware long term tracklet history construction. To select K images from the tracklet history of the target, we design a quality-aware mechanism. In order to get long term cues of good quality, we use a quality filter to select the best K images in the past K time periods to ensure quality and robustness. The indices of the K frames

Step	Branch A	Branch B	Branch C	Branch D
1	conv 1×1	conv 3×3	conv 3×3	conv 1×1
2	-	conv 1×1	conv 3×3	pool 3×3
3	-	-	conv 1×1	-

Table 2. One inception-A block: all convolution layers of 3×3 and the pooling layer use padding 1, the others have no padding, and the pooling layer stride 2.

selected as the tracklet history of the target are denoted by $\mathcal{H} = \{t_1 \ldots, t_K\}$. The K frames we choose are defined below:

$$t_i = \underset{t-i\delta<\hat{t}\leq t-(i-1)\delta}{\arg\max} \quad Q(I_{\hat{t}}^X), i = 1, 2, ..., K \tag{3}$$

where Q is a network which outputs the quality score. Q is implemented using a Resnet-18 model. I_t^X is the image region of target X at frame t. δ is hyper-parameter deciding the selecting interval. For example, when $i = 1$, t_1 is chosen from the frame with maximum quality score among frames $t, t-1, ..., t-\delta+1$. When $i = 2$, t_2 is chosen from $t-\delta, t-\delta-1, ..., t-2\delta+1$. Therefore, the i in t_i corresponds to different stride and search range.

Long-term feature generation. After the K images are selected, all these images and the detection result to be matched will be fed to the ReID sub-net and output their ReID features. Then we can obtain K long term features for the target as follows:

$$\begin{aligned} \mathcal{F}_l^X &= \{f_l(A_{t_i}^X A_{det}) | i = 1, \ldots, K\}, \\ \text{where } f_l(A_{t_i}^X, A_{det}) &= \frac{{A_{t_i}^X}^{\mathrm{T}} \cdot A_{det}}{|A_{t_i}^X| \, |A_{det}|}, \end{aligned} \tag{4}$$

$A_{t_i}^X$ is the vector containing ReID features of the ith image selected from tracklet history of target X, A_{det} is the ReID features of the detection result. To save computation, each image in the tracklet will have their features extracted by the ReID network only once. Their features are saved for further calculation.

3.4. Switcher-Aware Classifier

Switcher retrieval. We have observed large amount of identity switches (IDS) and find that most IDSs occur when two targets meet each other with large overlap. It inspires us to mark the other target having the largest overlap with current one as the most possible potential switcher. Mathematically, for each tracklet X in frame t, its position is denoted by X_t, and the potential switcher is obtained as follows

$$\Lambda = \underset{Y\in\mathcal{S}\ s.t.\ Y\neq X}{\arg\max} \quad IoU(X_t, Y_t). \tag{5}$$

where $\mathcal{S}$ is the set of tracked targets.

Input features. Here we consider the two sub-nets as a feature extraction operator ϕ, and denote the input of the two sub-nets for target X and detection result D as $\Gamma_{X,D}$, similarly for the switcher. The input features of the classifier consist of two parts: the features of mainly considered target, denoted by $\phi(\Gamma_{X,D})$, and the features of the switcher, denoted by $\phi(\Gamma_{\Lambda,D})$). ϕ is defined bellow.

$$\phi(\Gamma_{X,D}) = \{f_s(D_{track}, D_{det})\} \cup \mathcal{F}_l^X. \tag{6}$$

The dimension of $\phi(X, t)$ is $K + 1$, and similarly for the switcher. Then we obtain the input of the classifier by concatenating the two parts.

Classification. We exploit regularized Newton boosting decision tree with weighted quantile sketch, which is proposed by [8], in the classification step. If the classification result y is larger than threshold ζ_m, then the corresponding edge with cost $1 - y$ will be added to graph.

3.5. Training

3.5.1 Training data generation

The SOT sub-net and the ReID sub-net are trained independently. For the SOT sub-net, we generate some image pairs of targets according to the ground truth of the videos and the pairs are extended to include part of the background according to the training schema of Siamese-RPN. For better training, we only consider samples which have IoU to ground truth larger than 0.65 as positives and smaller than 0.45 as negatives. For ReID sub-net, each target is regarded as one class and we train the net to predict the class of the input target. The input of ReID sub-net is the target image region the label is its class number.

In order to generate training samples and corresponding annotations for the switcher-aware classifier, we should first generate the inputs of the two sub-nets. At the beginning, we run a baseline MOT algorithm and generate all hypothetical tracklets of training videos. Then we associate them with the ground truth using an IoU threshold of 0.6, i.e., only pairs with IoU larger than 0.6 will be considered. The sum of IoU value is maximized by Hungarian Algorithm. For target X at frame t, if the ground truth of $X_t, X_{t-\delta}, ..., X_{t-(K-1)\delta}$ belong to the same target or at most one of them is not associated with a ground truth, then we consider it as a valid tracklet. For each valid tracklet, we randomly sample a detection result in frame $t + 1$, together with the potential switcher defined by Eq. 5, the input of the MOT framework is done. According to Eq. 1, 4, 6, we can generate the inputs of the classifier. According to the ground truth of the detection, we can obtain the label of the switcher-aware classifier. Besides, for a positive sample, we can exchange the switcher and the mainly considered one to generate another training negative sample for the switcher-aware classifier.

As for the quality filter, we generate target image regions for input with IoU to ground truth larger than 0.6 as positive samples and the rest as negative samples.

3.5.2 Loss functions.

The loss function of the SOT sub-net is as the following:

$$L_{sot} = L_{sot}^{cls} + \lambda_{sot} L_{sot}^{reg}, \tag{7}$$

where λ_{sot} is a hyper-parameter for balancing the two loss functions. L_{sot}^{cls} is the cross-entropy classification loss and L_{sot}^{reg} is the regression loss.

We consider the re-identification task as a multi-class classification problem. A linear classifier is added after the backbone network and then the probability of each class is calculated through softmax. Finally we optimize the cross-entropy loss of the task:

$$L_{reid}(x, y) = -\sum_{i=0}^{N-1} y[i] \log(x[i]) \tag{8}$$

where N is the number of classes, x and y are network output and ground-truth, respectively.

For the Newton boosting tree classifier and the quality filter, we use the loss function $L_{cls}(x, y) = -(1 - y)\log(1 - x) - y\log(x)$.

3.6. Pre-Processing and Offline Clustering

Detection score filter strategy. Sometimes the detection results given by the detector are extremely noisy, with strange false positives and the confidences are unreliable. We propose two solutions to the refine these detection results. The first one is a stricter NMS method. The second one is to train a confidence score refiner. Here we use the quality filter in long term cues selection as the confidence refiner.

Long tracklets clustering Based on the output result of online mode, we design a simple batch clustering post-process procedure. For each tracklet, first we consider each frame as an isolated node, then if there are two nodes with similar ReID features, i.e., the cosine distance between them is less than a threshold, we will add an edge between the two nodes. Finally we can obtain some slices, each slice is one connected sub-graph. The second step is to merge these slices among different targets. Once again we merge two slices if they have small overlap in frame indexes, small spatial distance, and similar ReID features (we calculate mean feature distance of two slices). If two slices are merged, then the slices in the original place become new identities. Furthermore, after split and merge operations, we do interpolation in every tracklet in order to repair more false negatives.

4. Experiments

4.1. Implement Details

This framework is written in Python with PyTorch support. All CNNs are pre-trained on Imagenet dataset and then trained on MOT task. Additionally, we use extra private data in training two sub-net but only public data in training the classifier and the quality filter. The public data we use is 7 video sequences from MOT16 benchmark data. The private data contain labeled videos of pedestrians which are quite different from all testset videos. The amount of the private data for trainning is about 100 minutes of 25 fps videos and the number of different pedestrians is about 1000.

Tracklet status update. For matched tracklets, their positions will be updated as the position of the matched detection result and the SOT template will be updated. Otherwise, the position is kept unchanged, and so do the template. Despite the matching results, Kalman Filter is used to smooth the trajectories. The quality decay value is 0.95 and the exponent k = 16. When the overall quality score is lower than 0.5, the target will not be output. The matching threshold $\zeta_m = 0.05$ and drop threshold $\zeta_t = 0.1$.

Training. We use similar approach as the original work in the SOT sub-net but change anchors to $[1.0, 1.6, 2.2, 2.8, 3.4]$. When training the ReID sub-net, we use the Stochastic Gradient Descent (SGD) optimizer and set the initial learning rate to 0.1, decay by 0.5 every 8 epoches for 96 epoches in total. The ReID sub-net is trained with weight decay of 5×10^{-4} and mini-batch size of 32.

We train the swithcer aware classifier under xgboost framework[8]. The number of trees is set to 410. The max depth is 5, learning rate is 0.05 and the minimum leaf node weight is set to 1.

For long term cues, the selection interval of the tracklet history can be set from 10 to 20. We just simply sample $K = 3$ frames and set $\delta = 15$.

4.2. Evaluation on MOT Benchmarks

Datasets. The proposed framework is evaluated under the MOT16 and MOT17 benchmarks [26]. They share the same test videos but offer different detection input. MOT17 has fixed the ground truth and make them more accurate. The test video sequences include various complicated scenes and are still a great challenge.

Evaluate Metrics. Following the benchmarks, we evaluate our work with the CLEAR MOT Metrics[3]. Among the metrics, MOTA and IDF1 are regarded the most important. MOTA conclude the recall, precision as well as id-switch count. IDF1 indicates the average maximum consistent tracking rate. In evaluation of MOT trackers, the criteria MOTA is deeply relevant to the detector recall and precision, while ID-Switch impresses far less than them. How-

Benchmark	Method	**MOTA ↑**	MOTP ↑	**IDF1 ↑**	IDP ↑	IDR ↑	FP ↓	FN ↓	IDS ↓
MOT16	RAR16pub[12]	45.9%	74.8%	48.8%	69.7%	37.5%	6871	91173	648
MOT16	STAM16[10]	46.0%	74.9%	50.0%	71.5%	38.5%	6895	91117	473
MOT16	DMMOT[42]	46.1%	73.8%	54.8%	77.2%	42.5%	7909	89874	532
MOT16	AMIR[30]	47.2%	**75.8%**	46.3%	68.9%	34.8%	**2681**	92856	774
MOT16	MOTDT[24]	47.6%	74.8%	50.9%	69.2%	40.3%	9253	85431	792
MOT16	Ours	44.8%	75.1%	53.8%	75.2%	41.8%	9639	90571	**451**
MOT16	Ours(with filter)	**49.2%**	74.0%	**56.5%**	**77.5%**	**44.5%**	7187	**84875**	606
MOT17	PHD_GSDL17[13]	48.0%	**77.2%**	49.6%	68.4%	39.0%	23199	265954	3998
MOT17	AM_ADM17[21]	48.1%	76.7%	52.1%	71.4%	41.0%	25061	265495	2214
MOT17	DMAN[42]	48.2%	75.7%	55.7%	75.9%	44.0%	26218	263608	2194
MOT17	HAM_SADF17[39]	48.3%	**77.2%**	51.1%	71.2%	39.9%	**20967**	269038	1871
MOT17	MOTDT17[24]	50.9%	76.6%	52.7%	70.4%	42.1%	24069	250768	2474
MOT17	Ours	50.3%	76.8%	56.3%	**76.5%**	44.6%	21345	257062	**1815**
MOT17	Ours(with filter)	**52.7%**	76.2%	**57.9%**	76.3%	**46.6%**	22512	**241936**	2167
MOT16p	EAMTT_16[31]	52.5%	78.8%	53.3%	72.7%	42.1%	**4407**	81223	910
MOT16p	SORTwHPD16[5]	59.8%	**79.6%**	53.8%	65.2%	45.7%	8698	63245	1423
MOT16p	DeepSORT_2[36]	61.4%	79.1%	62.2%	72.1%	54.7%	12852	56668	781
MOT16p	RAR16wVGG[12]	63.0%	78.8%	63.8%	72.6%	56.9%	13663	53248	**482**
MOT16p	CNNMTT[25]	65.2%	78.4%	62.2%	73.7%	53.8%	6578	55896	946
MOT16p	POI[40]	66.1%	79.5%	65.1%	**77.7%**	56.0%	5061	55914	805
MOT16p	Ours	**69.6%**	78.5%	**68.6%**	77.1%	**61.7%**	9138	**45497**	768

Table 3. Comparision between the proposed MOT framework (online mode) with other online processing SOTA methods in MOT16 and MOT17. 'with filter' means detection score refiner is used. 'MOT16p' means MOT16 with private detection. Red for the best result.

Benchmark	Method	**MOTA ↑**	MOTP ↑	**IDF1 ↑**	IDP ↑	IDR ↑	FP ↓	FN ↓	IDS ↓
MOT17	IOU17[6]	45.5%	76.9%	39.4%	56.4%	30.3%	**19993**	281643	5988
MOT17	MHT_bLSTM[19]	47.5%	**77.5%**	51.9%	71.4%	40.8%	25981	268042	2069
MOT17	EDMT17[7]	50.0%	77.3%	51.3%	67.0%	41.5%	32279	247297	2264
MOT17	MHT_DAM_17[18]	50.7%	**77.5%**	47.2%	63.4%	37.6%	22875	252889	2314
MOT17	jCC[17]	51.2%	75.9%	54.5%	72.2%	43.8%	25937	247822	1802
MOT17	FWT_17[15]	51.3%	77.0%	47.6%	63.2%	38.1%	24101	247921	2648
MOT17	Ours(with filter)	**54.7%**	75.9%	**62.3%**	**79.7%**	**51.1%**	26091	**228434**	**1243**
MOT16p	NOMTwSDP16[9]	62.2%	79.6%	62.6%	77.2%	52.6%	**5119**	63352	406
MOT16p	MCMOT_HDM[20]	62.4%	78.3%	51.6%	60.7%	44.9%	9855	57257	1394
MOT16p	KDNT[40]	68.2%	79.4%	60.0%	66.9%	54.4%	11479	45605	933
MOT16p	LMP_p[35]	71.0%	**80.2%**	70.1%	78.9%	63.0%	7880	44564	**434**
MOT16p	HT_SJTUZTE[23]	**71.3%**	79.3%	67.6%	75.2%	61.4%	9238	42521	617
MOT16p	Ours	71.2%	78.3%	**73.1%**	**80.7%**	**66.8%**	10274	**41732**	510

Table 4. Comparision between the proposed MOT framework (batch mode) with other batch processing SOTA methods in MOT16 and MOT17. 'with filter' means detection score refiner is used. 'MOT16p' means MOT16 with private detection. Red for the best result.

ever IDF1 can indicate the consistency. A robust tracking system should have both high MOTA and IDF1 score.

Results. Table 3 and Table 4 show the results of online and batch processing methods both in MOT16 and MOT17, respectively. Besides, in task of MOT16 with private detector, our tracker, as well as KDNT, LMP_p, HT_SJTUZTE and POI trackers use the same detector that proposed by the author of POI tracker. The detections are available on google drive[1].

The results show that our framework outperforms many previous the state-of-the-art trackers in both MOT16 and MOT17 benchmarks. Both MOTA and IDF1 scores are in the leading position in MOT16/MOT17 among online/batch processing algorithms. Our batch processing algorithm achieved the highest MOTA in MOT17 benchmark[1].

[1]https://drive.google.com/open?id=0B5ACiy41McAHMjczS2p0dFg3emM

[1]Up to the date of 14/11/2018.

4.3. Ablation Study and Discussion

How do different cues influence the tracking quality? The ablation study was evaluated on MOT16 trainset. Since we have used the trainset for validation, we exclude the MOT16 trainset from the training data when training the sub-nets, the switcher-aware classifier and the quality filter for all ablation study results in Figure 4. For the classifier and quality filter, we use extra private training data in ablation study.

Figure 4 shows the impact of different components. The baseline model, (A) in Fig. 4, does not use a learned classifier but calculates affinity in a hand-crafted way using only

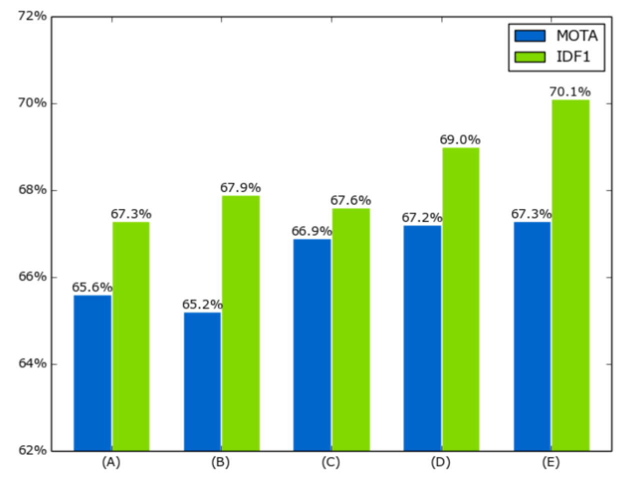


Figure 4. Analysis of our framework using different components. (A) baseline (hand-crafted) (B) long-term cues only (C) short-term cues only (D) long and short term cues (E) long and short term cues with SAC

position information. The other experimental results in Fig. 4 share the same settings of framework except the input features for classifier and four different classifiers are trained to fit the different input features. It can be seen from the figure that the short term cues provide more improvement for MOTA than long term cues when compared with the baseline. It is also the most intuitive reflection of the relevance between the two adjacent detection frames. When short term cues are utilized effectively, the MOTA score is improved by 1.3% and IDF1 is improved by 0.3 %. On the other hand, combining long term cues can effectively improve the discriminative ability between the tracklets, which brings increment of IDF1 by 1.4%. However, the MOTA improvement from long term cues is less than the improvement from the short term cues. Combining both short term and long term cues performs better than using single cue, which validates that these two cues are complementary to each other. Thirdly, adding the switcher-aware classifier can greatly reduce id-switches number, which leads to another 1.1% increment in IDF1, while it has just a little effect on MOTA. The learning approach to combine long term and short term cues is effective and the combination using SAC brings improvements on multiple metrics.

How does SAC work?

We also analyze the real effects on the videos. Figure 5 shows that with switcher-aware classification, the tracking is more robust. The main contribution of SAC is that it fixes a lot of id-switches. After SAC is used, in MOT16 private the id-switch number decreases from 642 to 569, IDF1 increases by 1.1%. This is because in traditional pairwise matching, lack of comparison to local switcher brings mistakes when the pair is occluded and therefore judged not

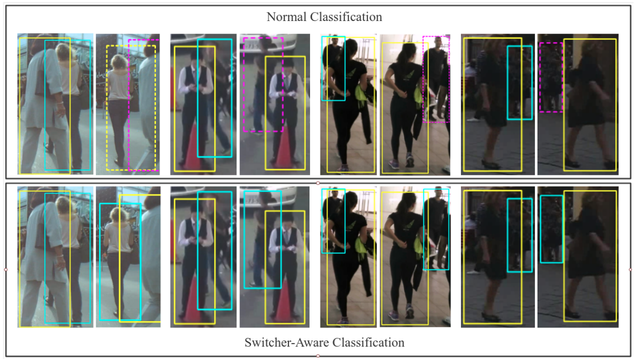


Figure 5. Examples of identity switch cases (top row) fixed when SAC is used (bottom row) for matching targets between adjacent frames. Dashed line boxes indicate id-switch.

to match. Besides, when occlusion happens, SAC helps to discriminate different targets.

Can multiple cues been handle in one network?

We have tried to extract some of the features from SOT backbone CNN and combine with the ReID branch through a ROI-Pooling layer. The experiment shows that a multi-task training of ReID and SOT task leads to drop in both SOT and ReID accuracy. Replacing Siamese-RPN by the multi-task trained network in MOT task, the MOTA decreases by 0.6% and IDF1 decreases by 2.5%. The SOT task needs background knowledge while the ReID task wants to eliminate it, which causes conflict during feature learning if a single network is used for both tasks. For the time being it is hard to handle two cues in one single network, further research works need to be done.

Why do we need small feature dimension?

We use a input feature of small dimension to balance the feature length of different information. It is well-known that motion and position features are usually very short. If we combine them with long appearance features, it is hard to fully utilize the position and motion features. We have tried directly concatenating raw ReID feature with short position features, the experimental result shows that imbalanced input features decreases IDF1 by 1.3%, MOTA by 0.1% and the IDS number increases by 88. We believe that the position and motion information are important and they should be emphasized by reducing the dimension of appearance features. Another inevitable issue is the complexity of data association. Short features make the procedure faster.

Why we use boosting trees for classification?

We have tried other different classifiers like neural network of linear layers(NN), support vector machine (SVM), but the boosting decision tree (BDT) is the best one. This experiment was done using full training data include MOT16. The MOTA of NN, SVM and BDT are 67.0%, 67.6% and 67.8%, respectively. We have found that for such a small input feature, neural network does not perform well on such small scale dataset.

5. Conclusions

In this paper, we have presented an effective MOT framework that learns to combine long term and short term cues. The long term cues can help correct the mistakes of short term cues, e.g., avoid the SOT sub-net to drift during occlusion. We also propose a switcher-aware classification method to improve the robustness of tracking system. The outstanding performance on MOT benchmarks demonstrates the effectiveness of the proposed framework.

References

[1] S.-H. Bae and K.-J. Yoon. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1218–1225, 2014.

[2] S.-H. Bae and K.-J. Yoon. Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):595–610, 2018.

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

[4] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr. Fully-convolutional siamese networks for object tracking. In *European conference on computer vision*, pages 850–865. Springer, 2016.

[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] J. Chen, H. Sheng, Y. Zhang, and Z. Xiong. Enhancing detection model for multiple hypothesis tracking. In *Conf. on Computer Vision and Pattern Recognition Workshops*, pages 2143–2152, 2017.

[8] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM, 2016.

[9] W. Choi. Near-online multi-target tracking with aggregated local flow descriptor. In *Proceedings of the IEEE international conference on computer vision*, pages 3029–3037, 2015.

[10] Q. Chu, W. Ouyang, H. Li, X. Wang, B. Liu, and N. Yu. Online multi-object tracking using cnn-based single object tracker with spatial-temporal attention mechanism. In *2017 IEEE International Conference on Computer Vision (ICCV).(Oct 2017)*, pages 4846–4855, 2017.

[11] M. Danelljan, G. Bhat, F. S. Khan, M. Felsberg, et al. Eco: Efficient convolution operators for tracking. In *CVPR*, page 3, 2017.

[12] K. Fang, Y. Xiang, X. Li, and S. Savarese. Recurrent autoregressive networks for online multi-object tracking. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 466–475. IEEE, 2018.

[13] Z. Fu, P. Feng, F. Angelini, J. Chambers, and S. M. Naqvi. Particle phd filter based multiple human tracking using online group-structured dictionary learning. *IEEE Access*, 6:14764–14778, 2018.

[14] D. Held, S. Thrun, and S. Savarese. Learning to track at 100 fps with deep regression networks. In *European Conference Computer Vision (ECCV)*, 2016.

[15] R. Henschel, L. Leal-Taixé, D. Cremers, and B. Rosenhahn. Fusion of head and full-body detectors for multi-object tracking. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018.

[16] C. Huang, B. Wu, and R. Nevatia. Robust object tracking by hierarchical association of detection responses. In *European Conference on Computer Vision*, pages 788–801. Springer, 2008.

[17] M. Keuper, S. Tang, B. Andres, T. Brox, and B. Schiele. Motion segmentation & multiple object tracking by correlation co-clustering. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[18] 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.

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

[20] B. Lee, E. Erdenee, S. Jin, M. Y. Nam, Y. G. Jung, and P. K. Rhee. Multi-class multi-object tracking using changing point detection. In *European Conference on Computer Vision*, pages 68–83. Springer, 2016.

[21] S.-H. Lee, M.-Y. Kim, and S.-H. Bae. Learning discriminative appearance models for online multi-object tracking with appearance discriminability measures. *IEEE Access*, Early Access:1–1, 2018.

[22] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu. High performance visual tracking with siamese region proposal network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8971–8980, 2018.

[23] W. Lin, J. Peng, S. Deng, M. Liu, X. Jia, and H. Xiong. Real-time multi-object tracking with hyper-plane matching. Technical report, Shanghai Jiao Tong University and ZTE Corp, 2017.

[24] C. Long, A. Haizhou, Z. Zijie, and S. Chong. Real-time multiple people tracking with deeply learned candidate selection and person re-identification. ICME, 2018.

[25] N. Mahmoudi, S. M. Ahadi, and M. Rahmati. Multi-target tracking using cnn-based features: Cnnmtt. *Multimedia Tools and Applications*, pages 1–20, 2018.

[26] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler. MOT16: A benchmark for multi-object tracking. *arXiv:1603.00831 [cs]*, Mar. 2016. arXiv: 1603.00831.

[27] A. Milan, S. H. Rezatofighi, A. R. Dick, I. D. Reid, and

K. Schindler. Online multi-target tracking using recurrent neural networks. In *AAAI*, volume 2, page 4, 2017.

[28] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial & Applied Mathematics*, 5(1):32–38, 1957.

[29] 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.

[30] A. Sadeghian, A. Alahi, and S. Savarese. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 300–311. IEEE, 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] J. Son, M. Baek, M. Cho, and B. Han. Multi-object tracking with quadruplet convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5620–5629, 2017.

[33] S. Tang, B. Andres, M. Andriluka, and B. Schiele. Subgraph decomposition for multi-target tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5033–5041, 2015.

[34] S. Tang, B. Andres, M. Andriluka, and B. Schiele. Multi-person tracking by multicut and deep matching. In *European Conference on Computer Vision*, pages 100–111. Springer, 2016.

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

[36] N. Wojke, A. Bewley, and D. Paulus. Simple online and realtime tracking with a deep association metric. In *Image Processing (ICIP), 2017 IEEE International Conference on*, pages 3645–3649. IEEE, 2017.

[37] Y. Xiang, A. Alahi, and S. Savarese. Learning to track: Online multi-object tracking by decision making. In *Proceedings of the IEEE international conference on computer vision*, pages 4705–4713, 2015.

[38] X. Yan, X. Wu, I. A. Kakadiaris, and S. K. Shah. To track or to detect? an ensemble framework for optimal selection. In *European Conference on Computer Vision*, pages 594–607. Springer, 2012.

[39] Y.-c. Yoon, A. Boragule, K. Yoon, and M. Jeon. Online multi-object tracking with historical appearance matching and scene adaptive detection filtering. *arXiv preprint arXiv:1805.10916*, 2018.

[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] 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.

[42] J. Zhu, H. Yang, N. Liu, M. Kim, W. Zhang, and M.-H. Yang. Online multi-object tracking with dual matching attention networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 366–382, 2018.