Online Multi-Object Tracking with Instance-Aware Tracker and Dynamic Model Refreshment

Peng Chu, Heng Fan, Chiu C Tan, and Haibin Ling
Temple University, Philadelphia, PA USA
{pchu, hengfan, chiu.tan, hbling}@temple.edu

Abstract

Recent progresses in model-free single object tracking *(SOT) algorithms have largely inspired applying SOT to* multi-object tracking *(MOT) to improve the robustness as well as relieving dependency on external detector. However, SOT algorithms are generally designed for distinguishing a target from its environment, and hence meet problems when a target is spatially mixed with similar objects as observed frequently in MOT. To address this issue, in this paper we propose an instance-aware tracker to integrate SOT techniques for MOT by encoding awareness both within and between target models. In particular, we construct each target model by fusing information for distinguishing target both from background and other instances (tracking targets). To conserve uniqueness of all target models, our instance-aware tracker considers response maps from all target models and assigns spatial locations exclusively to optimize the overall accuracy. Another contribution we make is a dynamic model refreshing strategy learned by a convolutional neural network. This strategy helps to eliminate initialization noise as well as to adapt to the variation of target size and appearance. To show the effectiveness of the proposed approach, it is evaluated on the popular MOT15 and MOT16 challenge benchmarks. On both benchmarks, our approach achieves the best overall performances in comparison with published results.*

1. Introduction

Tracking multiple objects in video is critical for many applications, ranging from vision-based surveillance to autonomous driving. A popular solution to Multiple Object Tracking (MOT) is the tracking-by-detection strategy, in which, detections from an external detector on each frame are associated and connected to form target trajectories in either online or offline batch mode. With recent progress on object detector, tracking-by-detection has been successful in multiple domains [1, 3, 7, 20, 27, 29, 30, 39, 44, 46]. However, separation of detection from tracking keeps detector inaccessible to the frame-to-frame correlation information which identifies the difference between object detection in still images and in videos. Moreover, the dependence on detection becomes a major limitation in complex scenes due to the degraded detection reliability caused by large size variation and partial occlusion of targets.

MOT, on the other hand, can be viewed as a generalized Single Object Tracking (SOT) problem where target locations are estimated from multiple SOT tracking models. Significant improvement has been achieved in recent SOT approaches which are efficient and robust in complex scenes [5, 14, 15, 17, 32]. However, even with a proper target management mechanism, directly applying multiple SOT trackers simultaneously to track multiple targets still experiences various difficulties.

A SOT tracker usually allows certain generalisability to capture appearance changes of target. In the MOT context, however, multiple similar targets may appear in the searching area of a SOT tracker. Such targets from the same category (e.g., pedestrian) often share similar appearance or shape that may confuse traditional SOT trackers. When this happens, SOT trackers for multiple targets may easily drift and even end up tracking the same target. Moreover, since SOT trackers depend heavily on the model learned at the first frame, steady tracking of a model-free SOT tracker requires groundtruth bounding box of target at the first frame to correctly distinguish the target from its background. In the current MOT framework, all target candidates are provided by a real detector which usually yields considerable noise in both target location and scale.

In this work, we propose using *instance-aware* (IA) tracker to both harvest the merit of SOT techniques and address the above issues in MOT. In addition to distinguishing a target from background as ordinary SOT tracker, our IA tracker tracks with the awareness of all other instances and their tracking models, which often means different targets of the same category. We implement such awareness in both target and global level. In scope of each target, we formulate the IA tracker in the efficient kernel correla-

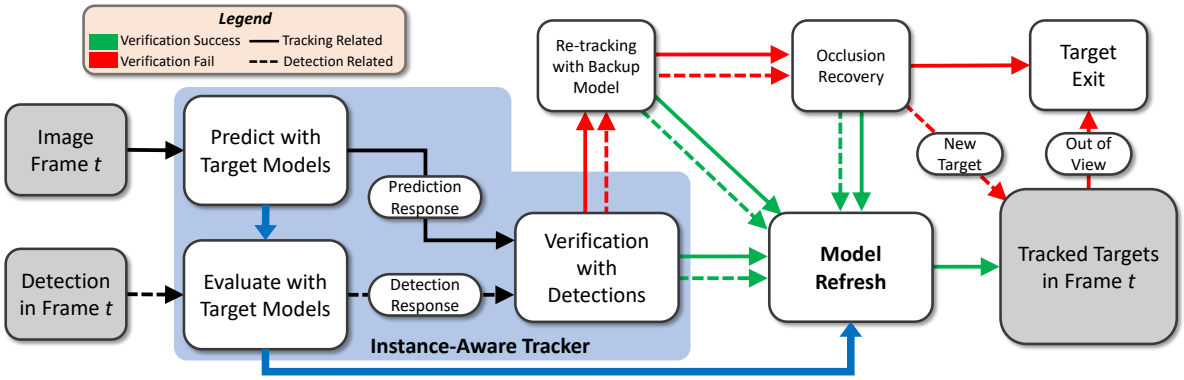

Figure 1. Overview of our instance-aware tracker based tracking system.

tion filter framework, while fusing features that tell a target from both background and other instances. This way, an IA target model is entitled with the awareness of differences between instances thus enhances response to its own target while suppressing responses to other similar targets. In global scope, generated response maps for all target models are used integrally to predict the target locations for a new coming frame. Awareness between targets models is treated as an optimization problem to maximize the overall response that each target is tracked exclusively by only one target model. A detection verification mechanism is proposed to solve the global optimization problem efficiently by incorporating detections from detectors and predictions from target models. And instead of updating model gradually, identity of a target model in proposed method is reinforced through a model refreshing mechanism, which is adaptively learned via a convolutional neural network.

Our contributions are mainly two-fold:

- We propose a novel instance-aware tracker to effectively integrate SOT in MOT. By being instance-aware inherently and mutually, target models significantly improve their capability to solve the ambiguity of similar targets in neighborhood.
- We propose an adaptive model refreshment strategy to further improve the reliability of SOT in MOT context.

To show the effectiveness of the proposed approach, it is evaluated on the popular MOT15 and MOT16 challenge benchmarks. On both benchmarks, our approach achieves the best overall performances in comparison with published results.

2. Related Work

Recent works on MOT primarily focuses on the tracking-by-detection principle. Most of these methods can be roughly categorized into two groups. The first group treat MOT as an offline global optimization problem that uses frame observation from both previous and future to estimate the current status of targets [2, 25, 34, 35, 40]. These methods usually focus on data association based methods such as Hungarian algorithm [6, 19], network flow [51, 52] and multiple hypotheses tracking [24]. Their performance heavily depends on the quality of detections from external detector. Different from these methods, our approach learns tracking model for each target to search and predict locations of next frame online. Detections in our approach are only used for model uniqueness verification and model refreshing.

The second group only needs observations till to the current frame to online estimate target status [10, 13, 42, 47–50]. In [47], MOT is formulated as a Markov decision process with a policy estimated on the labeled training data. [42] extends the work [47] to use deep CNN and LSTM to encode long-term temporal dependencies by fusing clues from motion, interaction and person re-identification model. Chu, et al. [10] use a dynamic CNN-based framework with a learned spatial-temporal attention map to handle occlusion, where CNN trained on ImageNet is used for pedestrian feature extraction. Yan et al. [48] gather target candidates from both detector and independent SOT trackers and select the optimal candidates through an ensemble model. Our approach differs from these methods by adding awareness between SOT trackers and dynamically refreshing model to eliminate possible noise in model initialization.

3. System Overview

For the t-th frame, our tracking system takes image frame and detections from an external detector as input, as shown in Fig. 1. Target models of instance-aware tracker are used to predict target locations independently and estimate scores for each detection. A detection verification process is applied to assign each spatial candidate exclusively to only one tracked target and verify the uniqueness of their target model as detailed in Sec. 4.2 and Sec. 4.3. Model of verified target will be refreshed if assigned detection enclosing target better than its model prediction. Unverified targets

and detections will be matched again using backup models to recover from incorrect refreshment. These components are explained in Sec. 4.4. Further, unpaired predictions and detections are passed into occlusion handling. Final unverified targets will exit when they have not been verified for some continuous frames. Unpaired detections will be added as new targets as described in Sec. 4.5.

4. Methodology

4.1. Problem Formulation

Following the tracking-by-detection paradigm, online MOT can be formulated as an optimization problem, at frame t, the set of N^t target locations $\hat{\boldsymbol{X}}^t = \{\hat{x}_i^t\}_{i=1}^{N^t}$ in current image I^t are chosen from M^t candidates in set $\boldsymbol{O}^t = \{x_j^t\}_{j=1}^{M^t}$ to maximize a score:

$$\hat{\boldsymbol{X}}^t = \arg\max_{\boldsymbol{X}^t \subset \boldsymbol{O}^t} f(I^t, \boldsymbol{X}^t; \mathbf{a}^t, \boldsymbol{W}^{t-1}). \quad (1)$$

$$\text{s.t.} \quad \sum_i a_{ij}^t \leq 1, a_{ij}^t \in \{0,1\}. \quad (2)$$

The parameter $\mathbf{a}^t = \{a_{ij}^t \in \{0,1\}\}$ indicates the association between the i-th tracked targets in $\hat{\boldsymbol{X}}^{t-1}$ at frame $t-1$ and the j-th candidate location in $\boldsymbol{X}^t$ at frame t. $a_{ij}^t = 1$ if $\hat{x}_i^{t-1}$ is associated with x_j^t, and $a_{ij}^t = 0$ otherwise. Each candidate can only be assigned to at most one tracked target. $\boldsymbol{W}^t = \{\mathbf{w}_i^t\}_{i=1}^{N^t}$ is the set of parameters to model each target, which is usually learned through a training procedure using the appearance or location information of target.

The objective function $f(\cdot)$ measures the overall quality of the tracking results for all targets at frame t, defined as below

$$f(I^t, \boldsymbol{X}^t) = \sum_{ij} a_{ij}^t g_i(I^t, x_j^t; \mathbf{w}_i^{t-1}). \quad (3)$$

The set of functions $g_i(.)$ can be interpreted as the objective function for tracking single target such that $g_i(I^t, x_j^t; \mathbf{w}_i^{t-1})$ assigns a score to the j-th candidate location x_j^t on I^t according to the i-th model parameter $\mathbf{w}_i^{t-1} \in \boldsymbol{W}^{t-1}$. The model parameters should be determined by previous images and target locations up to frame $t-1$.

Solving the online MOT problem, therefore, is to solve $\mathbf{a}^t$ and $g_i(.)$ for each frame.

4.2. Instance-Aware Tracker

We propose to use Instance-Aware (IA) tracker to solve $\mathbf{a}^t$ and $g_i(.)$ in two levels. For each single target, objective function $g_i(I^t, x_j^t)$ is learned to only assign a high score to its own target while returns low scores for both background and other instances. As for global, $\mathbf{a}^t$ is solved to associate each spatial location x_j^t on I^t exclusively to only one target referring those scores from all targets.

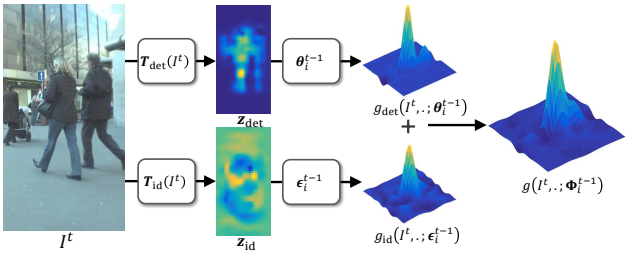

Figure 2. Illustration of target level instance-awareness: discriminate targets from background and discriminate different targets. $\mathbf{z}_{\text{det}}$ and $\mathbf{z}_{\text{id}}$ are feature maps visualized by accumulating values in all channels.

We start from the objective function $g_i(I^t, x_j^t)$. Ordinary SOT methods focus on distinguishing target from background and allow certain variations to handle appearance change of target, which makes tracker insensitive to distracters that are apparently similar to target. Thus, directly adopting SOT methods for MOT causes the trackers easily drifting to wrong targets. In this work, we treat the problem of tracking single target in MOT context as two subproblems: i) to distinguish targets from background; and ii) to model the difference between targets. The objective function can be rewritten as

$$g_i(I^t, x_j^t) = g_{\text{det}}(I^t, x_j^t; \boldsymbol{\theta}_i^{t-1}) + g_{\text{id}}(I^t, x_j^t; \boldsymbol{\epsilon}_i^{t-1}), \quad (4)$$

where $\boldsymbol{\theta}_i^{t-1}$ and $\boldsymbol{\epsilon}_i^{t-1}$ are the two model parameters for the i-th target focusing on each of the problems mentioned above, therefore $g_{\text{det}}(I^t, x_j^t; \boldsymbol{\theta}_i^{t-1})$ estimates the score for location x_j^t containing one of the targets using model parameter $\boldsymbol{\theta}_i^{t-1}$, $g_{\text{id}}(I^t, x_j^t; \boldsymbol{\epsilon}_i^{t-1})$ evaluates the similarity for object at x_j^t referring to the i-th target using model $\boldsymbol{\epsilon}_i^{t-1}$. Benefit of the separation is that for some target categories each of the sub-problems already has well-founded methods and datasets for model learning. For example, in the case of tracking multiple pedestrian, the first sub-problem is pedestrian detection and the second one is person re-identification, and both have large scale datasets such as MSCOCO, CUHK [31].

We focus on the Ridge Regression form of $g_{\text{det}}(.)$ and $g_{\text{id}}(.)$, in which, the functions share the form of $g(\mathbf{z}; \boldsymbol{\Phi}) = \boldsymbol{\Phi}\mathbf{z}^T$ with $\mathbf{z}$ for regression input and $\boldsymbol{\Phi}$ for learnable parameter. Then the objective function in Eq. 4 can be rewritten as

$$\begin{aligned} g_i(I^t, x_j^t) &= \boldsymbol{\theta}_i^{t-1} \boldsymbol{T}_{\text{det}}(I^t, x_j^t)^T + \boldsymbol{\epsilon}_i^{t-1} \boldsymbol{T}_{\text{id}}(I^t, x_j^t)^T \\ &= \boldsymbol{\Phi}_i^{t-1} \Big[\boldsymbol{T}_{\text{det}}(I^t, x_j^t), \boldsymbol{T}_{\text{id}}(I^t, x_j^t)\Big]^T \end{aligned} \quad (5)$$

where $\boldsymbol{T}_{\text{det}}(., x_j^t)$ and $\boldsymbol{T}_{\text{id}}(., x_j^t)$ are image transformations centering at x_j^t, $[.]$ is channel-wise concatenation, $\boldsymbol{\Phi}_i^{t-1}$ is

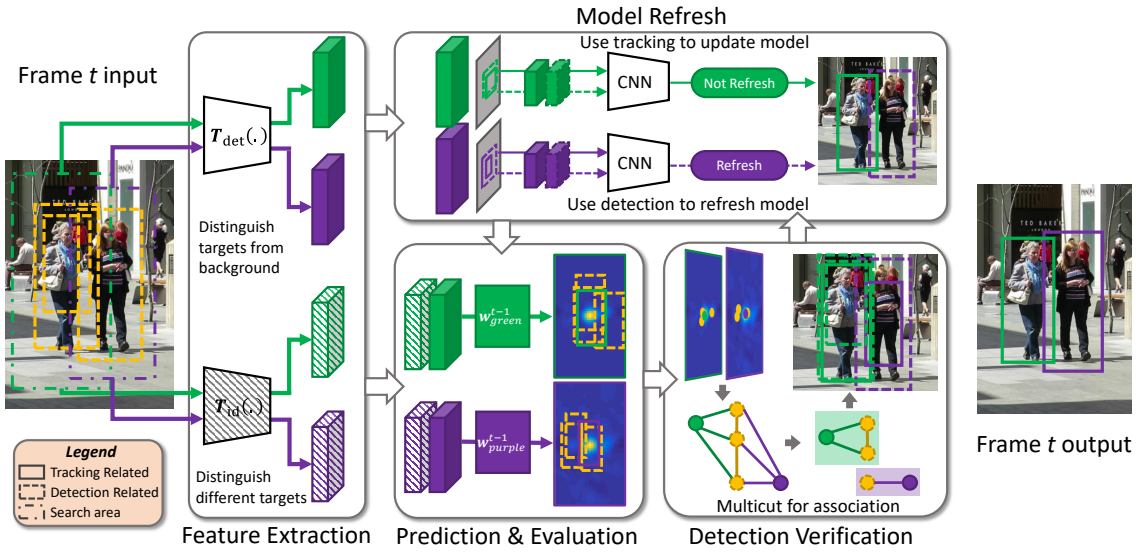

Figure 3. Tracking multiple objects by instance-aware tracker with model refreshment.

the combined model parameter for the i-th target. An illustration is shown in Fig. 2.

Solving $\mathbf{\Phi}_i^{t-1}$ online usually involves carefully designed strategies for positive and negative sample collection. Kernel Correlation Filter (KCF) tracker proposed in [17] solve this problem efficiently in Fourier domain by combining circulant matrices and kernel trick. Following this formulation, model parameter of the i-th target at frame t is obtained by $\tilde{\mathbf{w}}_i^{\prime t} = \frac{\tilde{\mathbf{y}}}{\tilde{\mathbf{k}}^{\mathbf{z}^t\mathbf{z}^t}+\lambda}$, where $\mathbf{z}_t = \left[\boldsymbol{T}_{\text{det}}(I^t), \boldsymbol{T}_{\text{id}}(I^t)\right]$ is the feature map, $\mathbf{k}^{\mathbf{z}^t\mathbf{z}^t}$ is defined as kernel correlation in [17], $\tilde{\mathbf{y}} = \mathcal{F}(\mathbf{y})$ is Discrete Fourier Transform (DFT) of regression labels. If considering a SOT context where only one target presents, an predicted location ${}^P\hat{x}_i^t$ using the i-th target model then can be estimated as

$$ {}^P\hat{x}_i^t = \underset{x_j^t \in \boldsymbol{O}^t}{\arg\max}\, \mathcal{F}^{-1}(\tilde{\mathbf{k}}^{\mathbf{z}^{t-1}\mathbf{z}^t} \odot \tilde{\mathbf{w}}_i^{t-1})\big|_{=x_j^t}. \tag{6} $$

Now we apply the objective function of tracking a single target to MOT context by combining Eq. 3 and Eq. 6. Given the set of $\{x_j^t\}$, the objective function of IA tracker subject to constrain in Eq. 2 is defined as

$$ f(I^t, \boldsymbol{X}^t) = \max_{a_{ij}} \sum_{ij} a_{ij}^t \mathcal{F}^{-1}(\tilde{\mathbf{k}}^{\mathbf{z}^{t-1}\mathbf{z}^t} \odot \tilde{\mathbf{w}}_i^{t-1})\big|_{=x_j^t} \tag{7} $$

The core idea of IA tracker can be explained as follow. In SOT version of KCF tracker, prediction is the spatial location strongest responding on response map given by $\mathcal{F}^{-1}(\tilde{\mathbf{k}}^{\mathbf{z}^{t-1}\mathbf{z}^t} \odot \tilde{\mathbf{w}}_i^{t-1})$. While in MOT, each spatial location has multiple responses generated by different target models of frame $t-1$ as shown in Fig. 3. And to further confirm each target is tracked exclusively by only one target model, we make use of the spatial exclusive assumption that no two or more targets can occupy the same position on image at the same time (on image frame only, not considering real 3D space). Thus, a global optimization in Eq. 7 is employed to maximize the overall response subject to the spatial exclusive constrain defined in Eq. 2 that each spatial location on image belongs to at most one target. Notice that, in calculation, $\mathbf{z}^t$ usually covers the search area of a target model only, where it should be written as $\mathbf{z}_i^t$ and the actual coordinate of x_j^t in $\mathbf{z}_i^t$ should also be converted accordingly.

4.3. Detection Verification

Solving the optimization problem in Eq. 7 for all spatial locations on image frame, e.g. each pixel, is computationally impractical. Ideally, a subset whose elements are complete and spatial exclusive is preferred. Prediction ${}^P x_j^t \in \boldsymbol{P}^t$ from Eq. 6 contains all possible locations for all targets, but these locations may have potential spatial conflicts. Detections ${}^D x_j^t \in \boldsymbol{D}^t$ from a category detector are spatial exclusive but not complete due to the possible false negative. We use the combination of detection ${}^D x_j^t \in \mathbf{D}^t$ and predictions ${}^P x_j^t \in \mathbf{P}^t$ as the candidate locations set $\boldsymbol{O}^t = \boldsymbol{D}^t \cup \boldsymbol{P}^t$. Result candidate set is complete but only partially spatial exclusive. Therefore, we propose a detection verification mechanism to solve $\mathbf{a}^t$ for all targets leveraging the limited spatial exclusive information provided by $\boldsymbol{D}^t$.

If a graph $G(\boldsymbol{V}, \boldsymbol{E})$ is created on $\boldsymbol{V} = \hat{\boldsymbol{X}}^{t-1} \cup \boldsymbol{O}^t$ and $\boldsymbol{E}$ which are the edges between vertexes in $\boldsymbol{V}$, the optimization problem in Eq. 7 with constrain in Eq. 2 can be reformed as a graph multicut problem minimizing cost:

$$ \min_{c_e \in \{0,1\}} \sum_{e \in E} c_e d_e \tag{8} $$

$$\text{s.t.} \quad \forall P_{uv} \in \mathcal{P} \quad \forall e \in P_{uv} : c_e \leq \sum_{e' \in P_e / \{e\}} c_{e'}, \qquad (9)$$

where $u, v \in \boldsymbol{V}$, c_e is the binary label indicating if $e \in \boldsymbol{E}$ is a cutting edge, d_e is the cost/reward associated to edge e, P_{uv} is the set of path from u to v, e is the edge between u and v. Solving Eq. 8 and Eq. 9 in the context of MOT is to find the subgraphs, in which, candidate locations belonging to the same target are connected while belonging to different targets are separated by cutting edges as shown in Fig. 3.

After optimization, each ${}^{D}x_j^t$ is assigned to one of the tracked target $\hat{x}_i^{t-1} \in \hat{\boldsymbol{X}}^{t-1}$. Verification of each target tracked exclusively by only one tracking models can be done by checking whether a tracked target assigned with detections. Due to the possible false negative and false positive generated by a real detector, verification can only be conducted every T_V frames to confirm the uniqueness of target model in long-term. Particular, if a tracked target has not been assigned with any detection for continuous T_V frames, then its target model is likely tracking either a false positive target or a target shared with other models. Increasing T_V, therefore, decreases awareness between target models since it allows each target model to track independently for more frames. T_V also controls the dependency on external detection and can be adjusted to adapt different detection qualities. Detailed parameter choice and discuss for T_V are described in Sec. 5.2.

We employ a primal heuristic based approach proposed in [21] for solving Eq. 8 with Eq. 9, where a set of transformation sequences are used to update the bi-partitions of a subgraph. Specifically, cost of each edge in Eq. 8 is calculated as:

$$d_e \doteq d_{uv} = \begin{cases} g_i(I^t, x_j^t), & \text{if } u \in \hat{\boldsymbol{X}}^{t-1}, v \in \boldsymbol{O}^t \\ \text{IoU}(\boldsymbol{b}_j^t, \boldsymbol{b}_{j'}^t), & \text{if } u \in \boldsymbol{O}^t, v \in \boldsymbol{O}^t \\ -C, & \text{if } u \in \hat{\boldsymbol{X}}^{t-1}, v \in \hat{\boldsymbol{X}}^{t-1} \end{cases} \qquad (10)$$

where $\text{IoU}(.)$ calculates the bounding box overlap ratio in term of Intersection over Union, $\boldsymbol{b}_j^t$ is the bounding box associated with x_j^t, and C is large positive constant to ensure cutting between different tracked targets. The final equivalence between a_{ij} and c_e is defined as following

$$a_{ij} = \begin{cases} \bar{c}_e = \bar{c}_{uv}, & \text{if } u \in \hat{\boldsymbol{X}}^{t-1}, v \in \boldsymbol{O}^t \\ 0, & \text{otherwise} \end{cases} \qquad (11)$$

where $\bar{c}_e$ stands for the logical negation.

4.4. Model Refresh

We train a CNN based classifier to determine whether to refresh the tracking model of a target using its assigned detection. Target model in ordinary SOT methods is initialized by target groundtruth in the first frame and is slowly and constantly updated. While in MOT, models are initially learned from detections which contain considerable noise in location and scale. And when targets moving close to the camera, their scale also will change rapidly. Due to those reasons, models in MOT have to be refreshed frequently.

Specifically, feature maps centering at tracked target ${}^{P}\hat{x}_i^t$ and its assigned detection ${}^{D}x_j^t$ are extracted and stacked channel-wise to feed into a CNN based classifier. The CNN is to make comparison between the bounding boxes associated with ${}^{P}\hat{x}_i^t$ and ${}^{D}x_j^t$ on target enclosing. If the bounding box of ${}^{D}x_i^t$ encloses target better, $\mathbf{w}_i^t$ will be refreshed by re-calculating $\mathbf{w}_i^t$ using ${}^{D}x_i^t$. In tracking phase, we reuse features from $\boldsymbol{T}_{\text{det}}(I^t)$ and adopt ROI Pooing to exact feature maps at specific locations, as show in Fig. 3

We adopt reinforcement learning to train the CNN classifier for the model refreshing policy. We update the classifier or the policy only when it makes a mistake. Suppose the tracker is tracking the i-th target in t-th frame. There are two types of mistake that can happen. i) Bounding box of ${}^{D}x_i^t$ encloses target better than ${}^{P}\hat{x}_i^t$ referring to groundtruth bounding box, but classifier chooses not to refresh $\mathbf{w}_i^t$. Then features at ${}^{D}x_i^t$ and ${}^{P}\hat{x}_i^t$ are concatenated and added to training set as positive samples. ii) Bounding box of ${}^{P}\hat{x}_i^t$ encloses target better than ${}^{D}x_i^t$, but classifier chooses to refresh $\mathbf{w}_i^t$. Concatenated features in those cases are added as negative samples. Each time the classifier makes a mistake, the CNN is trained through a constant number of iterations using online batches of size B_N, which contains the newly added sample and $B_N - 1$ samples randomly sampled from the rest training set. We keep updating the policy until all the targets in training set are successfully tracked.

In case of classifier making mistake at real tracking phase, we adopt a model backup mechanism. In frame t, if classifier chooses to refresh $\mathbf{w}_i^{t-1}$ with a new $\mathbf{w}_i^t$, $\mathbf{w}_i^{t-1}$ will be saved. In frame $t+1$, if tracker with $\mathbf{w}_i^t$ cannot be assigned with a ${}^{D}x_j^{t+1}$, the old model $\mathbf{w}_i^{t-1}$ will be restored for tracking and verification one more time.

4.5. Target Management

In this work, except for 'Tracked' event, we also handle the 'Occlusion', 'Enter' and 'Exit' events of targets.

Occlusion To recovery a target from occlusion, we train an SVM classifier to estimate if two locations $\hat{x}_i^{t'}$ and ${}^{D}x_j^t$ are containing the same target. We make a simple assumption that a detection ${}^{D}x_j^t$ not assigned to any tracked target in detection verification and re-tracking phase is either a new target or an existing target just finished occlusion. Occlusion recovery thus is to connect that detection with tracked target not assigned with detection. Suppose i-th target starts to be occluded in frame t' and finishes occlusion in frame t, where $t - t' > 1$, $\mathbf{b_1} = (\xi_1, \zeta_1, \omega_1, \eta_1)$ is the bounding box associated with the first location specified by its x-coordinate, y-coordinate, width and height respectively,

Table 1. Tracking Performance on the MOT training set.

Venice-2											
T_V	0	1	2	3	4	5	6	7	8	10	20
MOTA	27.0	30.9	32.7	33.7	34.4	32.4	33.1	33.4	33.1	34.1	31.5
FP	647	708	773	795	824	864	889	922	942	961	1237
FN	4498	4173	3991	3898	3825	3931	3856	3801	3801	3714	3622
MOT16-05											
T_V	0	1	2	3	4	5	6	7	8	10	20
MOTA	39.5	40.3	40.9	41.1	41.2	42.3	42.3	42.2	40.9	40.0	37.8
FP	187	231	255	281	312	324	340	355	415	492	693
FN	3522	3441	3387	3349	3314	3240	3228	3216	3241	3222	3152

Table 2. Tracking Performance on the MOT training set

Method	MOTA↑	MOTP↑	MT↑	ML↓	FP↓	FN↓	IDS↓	Frag↓
IA	26.1	69.3	15.4%	26.9%	991	4250	39	42
IA+MR	33.3	74.0	15.4%	34.6%	785	3939	36	53
IA−DV+MR	-35.3	72.7	38.5%	19.3%	6734	2856	70	108
IA−TA+MR	32.2	73.7	15.4%	38.5%	760	4039	43	53
Full	34.4	74.1	15.4%	30.8%	824	3825	36	66

and $\mathbf{b}_2$ is for the second location. We can calculate the following feature for estimation,

$$\left[t-t', \frac{\xi_2-\xi_1}{\bar{\eta}}, \frac{\zeta_2-\zeta_1}{\bar{\eta}}, \frac{\eta_2-\eta_1}{\bar{\eta}}, \mathrm{IoU}(\mathbf{b}_2, \mathbf{b}_1), \phi_{hist}\right],$$

where $\bar{\eta} = \frac{\eta_1+\eta_2}{2}$ and ϕ_{hist} is histogram intersection of the two image patches bounded by $\mathbf{b}_1$ and $\mathbf{b}_2$. In the tracking phase, for those ${}^{D}x_j^t$ not assigned to any $\hat{x}_i^{t-1}$ and those $\hat{x}_i^{t'}$ not being assigned with any ${}^{D}x_j^t$, the SVM classifier is used to estimate the matching possibility of each pair. Hungarian algorithm is employed to find the final matching pair.

Target Enter As mentioned above, if ${}^{D}x_j^t$ hasn't been assigned in any of the previous stages, ${}^{D}x_j^t$ is added to $\hat{\boldsymbol{X}}_t$ as a new target.

Target Exit We adopt two criteria for target exit checking: i) Bounding box of $\hat{x}_i^t$ is out of view. ii) Target hasn't been assigned a detection for continuous T_V frames.

5. Experiments

We conduct three experiments on the popular MOT15 [28] and MOT16 [36] benchmarks to analyze our proposed approach and compare to prior works. The test set of MOT15 contains 11 sequences and MOT16 contains 7 sequences, where camera motion, camera angle, and imaging condition vary greatly. For each test sequence, a training sequence is provided which is captured in the similar settings. For both training and test set, detections from a real detector are provided.

5.1. Experiment Setting

The proposed approach is implemented in MATLAB with Caffe and running on a desktop with 4 cores@3.60GHz CPU and a GTX1080 GPU. We use PAFNet proposed in [8] for $\boldsymbol{T}_{\mathrm{det}}(.)$. PAFNet generates two feature maps at the end, where different human body parts and corresponding affinity field are highly responded. Two feature maps are concatenated along channel to form the output of $\boldsymbol{T}_{\mathrm{det}}(.)$. We use $\boldsymbol{T}_{\mathrm{det}}(.)$ to distinguish pedestrian from their background. PartNet proposed in [53] is adopt for $\boldsymbol{T}_{\mathrm{id}}(.)$. PartNet generate $L2$ normalized feature for person Re-Identification task, which is suitable for $\boldsymbol{T}_{\mathrm{id}}(.)$ to distinguish different pedestrians. Original PartNet outputs a feature vector for each input image. We remove its last global pooling layer and convert the last fully connected (FC) layer to convolutional layer to output feature map in reasonable dimensions.

For each test sequence in MOT15 and MOT16 dataset, one or more similar sequences in training set are used to train a CNN classifier mentioned in Sec. 4.4 and a SVM classifier in Sec. 4.5. We adopt the partition method mentioned in [47]. CNN classifier is consisted of one convolutional layer and one FC layer. When training the CNN classifier, $B_N = 32$ and 5 iterations with constant learning rate of 0.001 are used when CNN classifier makes mistake.

By implementing IA tracker and model refreshment with shared feature extraction as shown in Fig. 3, the average speed of proposed approach on MOT15 dataset is about 0.3 fps and 0.1 fps on MOT16 dataset. The average target densities on each frame of those two datasets are 10.6 and 30.8 for the test set. Proposed method achieves acceptable speed

Table 3. Tracking Performance on the MOT15 benchmark test set. Best in bold.

Mode	Method	MOTA↑	MOTP↑	MT↑	ML↓	FP↓	FN↓	IDS↓	Frag↓
Offline	TBD [16]	15.9	70.9	6.4%	47.9%	14943	34777	1939	1963
	CEM [38]	19.3	70.7	8.5%	46.5%	14180	34591	813	1023
	JPDA_m [41]	23.8	68.2	5.0%	58.1%	**4533**	41873	**404**	**792**
	SiameseCNN [26]	29.0	71.2	8.5%	48.4%	5160	37798	639	1316
	MHT_DAM [24]	32.4	71.8	16.0%	43.8%	9064	32060	435	826
	JMC [22]	**35.6**	**71.9**	**23.2%**	**39.3%**	10580	**28508**	457	969
Online	RNN [37]	19.0	71.0	5.5%	45.6%	11578	36706	1490	2081
	oICF [23]	27.1	70.0	6.4%	48.7%	7594	36757	**454**	1660
	SCEA [18]	29.1	71.1	8.9%	47.3%	6060	36912	604	**1182**
	MDP [47]	30.3	71.3	13.0%	38.4%	9717	32422	680	1500
	AP [33]	38.5	**72.6**	8.7 %	37.4%	**4005**	33203	586	1263
	proposed	**38.9**	70.6	**16.6%**	**31.5%**	7321	**29501**	720	1440

Table 4. Tracking Performance on the MOT16 benchmark test set. Best in bold.

Mode	Method	MOTA↑	MOTP↑	MT↑	ML↓	FP↓	FN↓	IDS↓	Frag↓
Offline	SMOT [12]	29.7	75.2	5.3%	47.7%	17426	107552	3108	4483
	CEM [38]	33.2	75.8	7.8%	54.4%	6837	114322	642	731
	GMMCP [11]	38.1	75.8	8.6%	50.9%	6607	105315	937	1669
	MHT_DAM [24]	45.8	76.3	16.2%	43.2%	**6412**	91758	590	781
	NOMT [9]	46.4	76.6	**18.3%**	41.4%	9753	87565	**359**	**504**
	LMP [45]	**48.8**	**79.0**	18.2%	**40.1%**	6654	**86245**	481	595
Online	OVBT [2]	38.4	75.4	7.5%	47.3%	11517	99463	1321	2140
	EAMTT [43]	38.8	75.1	7.9%	49.1%	8114	102452	965	1657
	oICF [23]	43.2	74.3	11.3%	48.5%	6651	96515	**381**	1404
	AMIR [42]	47.2	**75.8**	14.0%	41.6%	**2681**	92856	774	1675
	proposed	**48.8**	75.7	**15.8%**	**38.1%**	5875	**86567**	906	**1116**

performance compared with other methods such as LMP (offline) [45] at 0.6 fps and AMIR (online) [42] at 1.0 fps.

Evaluation Metric To evaluate the performance of proposed method, we employ the widely accepted CLEAR MOT metrics [4], including multiple object tracking precision (MOTP) and multiple object tracking accuracy (MOTA) which is a cumulative measure that combines false positives (FP), false negatives (FN) and the identity switches (IDS). Additionally, we also report the percentage of mostly tracked targets (MT), the percentage of mostly lost targets (ML), and the number of times a trajectory is fragmented (Frag).

5.2. Determine T_V

Hyper-parameter T_V in proposed approach is used to determine the maximum continuous frames that a target can be tracked without the verification from external detection. Setting of T_V controls the strength of awareness between target models: As T_V increasing, verification becomes less frequent, each target model tracks its target more independently, which is more equivalent with directly applying multiple SOT tracker for MOT; When decreasing T_V, proposed approach depends more on external detection and behaviors more like traditional tracking-by-detection approaches. As reflected in evaluation metrics, choice of T_V controls the trade off between FP and FN. Higher T_V allows tracker to continue more frames without the confirmation from detection, thus may introduce more FP. Lower T_V requires frequent verification between tracker and detection, where tracking performance will heavily depend on detection quality, thus tracker may generate more FN when detection quality gets worse.

We test various of T_V on the training dataset of MOT benchmark. The results of MOTA, FP and FN for Venice-2 from MOT15 and MOT16-05 from MOT16 are reported in Tab. 1. In both sequences, starting at $T_V = 0$ where verification for every frame is required and increasing T_V, MOTA first increases then decreases due to the increasing FP in results. FP and FN gain their balance at $T_V = 4$ for Venice-2 and $T_V = 5$ for MOT16-05 where MOTA achieves best. We choose $T_V = 4$ for the rest of our experiments.

5.3. Ablation Study

We justify the effectiveness of each building block in proposed method through ablation study as shown in Tab. 2.

Figure 4. Visualization of selected sequences. The first row is from MOT15 test set, the second row is from MOT16 test set. Trajectories are fitted for better view.

IA stands for the proposed instance-aware tracker. MR is the dynamic model refreshing. IA−DV disables detection verification in IA tracker by setting $T_V \rightarrow \infty$, which removes the awareness between target models thus is equivalent with applying multiple independent SOT trackers for MOT. IA−TA disables target level awareness by replacing the fusion features with general deep features extracted from VGG-16 trained on ImageNet. Full method also includes the re-tracking and occlusion handle part.

Analysis is performed on Venice-2 sequence from training set of MOT15. Numerical results of all CLEAR MOT metrics are listed in Tab. 2. Having demonstrated the importance of awareness between target models in Sec. 5.2, totally disabling detection verification results in the greatest performance degradation. Model refreshment is also essential for improving performance and robust tracking. As shown by MOTA and MOTP, with model refreshment, not only tracking accuracy but also the bounding box precision improves a lot. In Full method, re-tracking and occlusion handle mechanism use simple linear interpolation to estimate the missing locations between the previous tracked target and current detection, which may introduce FP, but reduces more FN as shown in Tab. 2 thus still improves the overall performance.

5.4. Results on Test Sequences

We test our proposed approach on both MOT15 and MOT16 test sequences. In order to boost performance, we adopt several pre- and post- processing techniques, including excluding detections with extreme size according to scene prior and applying fitting to result trajectories in sequences where no rapid pedestrian scale changes. The performance is shown in Tab. 3 and Tab. 4. We compared our method with the best peer-reviewed and published results on the benchmark, including JMC [22], AP [33], LMP [45] and AMIR [42].

The biggest challenge in MOT15 and MOT16 datasets is the enormous FN over FP (more than 10 times in MOT16) as shown in Tab. 3 and Tab. 4, which is partially introduced by FN in public detection. Benefited from the built-in SOT techniques, proposed method results in the least number of FN and the best MT/ML performance compared with all other online methods. As for overall performance, we established a new state-of-the-art among all online and offline methods in both MOT15 and MOT16 benchmark in terms of MOTA which is the most important metric for MOT. Visualization of selected sequences is shown in Fig. 4. The complete metrics and visualization can be found on the benchmark website.[1]

6. Conclusion

In this paper we proposed using instance-aware with SOT technique to improve multiple object tracking (MOT). By built-in instance-awareness both in each target model and between all target models, our proposed approach can better predict the location of each target online, and meanwhile conserves the uniqueness of each tracking model to prevent the generation of duplicated and false positive trajectory. Tracking models in our approach are refreshed dynamically with a learned convolutional neural network to inhibit the noise of using inaccurate detections and to adapt appearance and scale variation of targets over time. Experiments on the MOT15 and MOT16 challenge datasets show the effectiveness of proposed approach in comparison with state-of-the-art.

Acknowledgement. This work is supported in part by US NSF Grants 1407156, 1618398 and 1814745.

[1] https://motchallenge.net/results/2D_MOT_2015/ and https://motchallenge.net/results/MOT16/ referred as 'KCF'.

References

[1] S.-H. Bae and K.-J. Yoon. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In *CVPR*, 2014. 1

[2] Y. Ban, S. Ba, X. Alameda-Pineda, and R. Horaud. Tracking multiple persons based on a variational bayesian model. In *ECCV*, 2016. 2, 7

[3] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *TPAMI*, 33(9):1806–1819, 2011. 1

[4] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *JIVP*, 2008:1, 2008. 7

[5] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. Torr. Staple: Complementary learners for real-time tracking. In *CVPR*, 2016. 1

[6] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft. Simple online and realtime tracking. In *ICIP*, 2016. 2

[7] W. Brendel, M. Amer, and S. Todorovic. Multiobject tracking as maximum weight independent set. In *CVPR*, 2011. 1

[8] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017. 6

[9] W. Choi. Near-online multi-target tracking with aggregated local flow descriptor. In *ICCV*, 2015. 7

[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 *ICCV*, 2017. 2

[11] A. Dehghan, S. Modiri Assari, and M. Shah. Gmmcp tracker: Globally optimal generalized maximum multi clique problem for multiple object tracking. In *CVPR*, 2015. 7

[12] C. Dicle, O. I. Camps, and M. Sznaier. The way they move: Tracking multiple targets with similar appearance. In *ICCV*, 2013. 7

[13] L. Fagot-Bouquet, R. Audigier, Y. Dhome, and F. Lerasle. Online multi-person tracking based on global sparse collaborative representations. In *ICIP*, 2015. 2

[14] H. Fan and H. Ling. Parallel tracking and verifying: A framework for real-time and high accuracy visual tracking. In *ICCV*, 2017. 1

[15] H. Fan and H. Ling. Sanet: Structure-aware network for visual tracking. In *CVPRW*, pages 2217–2224. IEEE, 2017. 1

[16] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun. 3d traffic scene understanding from movable platforms. *TPAMI*, 36(5):1012–1025, 2014. 7

[17] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *TPAMI*, 37(3):583–596, 2015. 1, 4

[18] J. Hong Yoon, C.-R. Lee, M.-H. Yang, and K.-J. Yoon. Online multi-object tracking via structural constraint event aggregation. In *CVPR*, 2016. 7

[19] C. Huang, B. Wu, and R. Nevatia. Robust object tracking by hierarchical association of detection responses. In *ECCV*, 2008. 2

[20] H. Jiang, S. Fels, and J. J. Little. A linear programming approach for multiple object tracking. In *CVPR*, 2007. 1

[21] M. Keuper, E. Levinkov, N. Bonneel, G. Lavoué, T. Brox, and B. Andres. Efficient decomposition of image and mesh graphs by lifted multicuts. In *ICCV*, 2015. 5

[22] M. Keuper, S. Tang, Y. Zhongjie, B. Andres, T. Brox, and B. Schiele. A multi-cut formulation for joint segmentation and tracking of multiple objects. *arXiv preprint arXiv:1607.06317*, 2016. 7, 8

[23] H. Kieritz, S. Becker, W. Hübner, and M. Arens. Online multi-person tracking using integral channel features. In *AVSS*, 2016. 7

[24] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg. Multiple hypothesis tracking revisited. In *ICCV*, 2015. 2, 7

[25] H.-U. Kim and C.-S. Kim. Cdt: Cooperative detection and tracking for tracing multiple objects in video sequences. In *ECCV*, 2016. 2

[26] L. Leal-Taixé, C. Canton-Ferrer, and K. Schindler. Learning by tracking: Siamese cnn for robust target association. In *CVPRW*, 2016. 7

[27] L. Leal-Taixé, M. Fenzi, A. Kuznetsova, B. Rosenhahn, and S. Savarese. Learning an image-based motion context for multiple people tracking. In *CVPR*, 2014. 1

[28] 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. 6

[29] B. Leibe, K. Schindler, and L. Van Gool. Coupled detection and trajectory estimation for multi-object tracking. In *ICCV*, 2007. 1

[30] P. Lenz, A. Geiger, and R. Urtasun. Followme: Efficient online min-cost flow tracking with bounded memory and computation. In *ICCV*, 2015. 1

[31] W. Li, R. Zhao, T. Xiao, and X. Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *CVPR*, 2014. 3

[32] Y. Li and J. Zhu. A scale adaptive kernel correlation filter tracker with feature integration. In *ECCV*, 2014. 1

[33] C. Long, A. Haizhou, Z. Chong, S.and Zijie, and B. Bo. Online multi-object tracking with convolutional neural networks. In *ICIP*, 2017. 7, 8

[34] S. Manen, R. Timofte, D. Dai, and L. Van Gool. Leveraging single for multi-target tracking using a novel trajectory overlap affinity measure. In *WACV*, 2016. 2

[35] N. McLaughlin, J. M. Del Rincon, and P. Miller. Enhancing linear programming with motion modeling for multi-target tracking. In *WACV*, 2015. 2

[36] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016. 6

[37] A. Milan, S. H. Rezatofighi, A. R. Dick, I. D. Reid, and K. Schindler. Online multi-target tracking using recurrent neural networks. In *AAAI*, 2017. 7

[38] A. Milan, S. Roth, and K. Schindler. Continuous energy minimization for multitarget tracking. *TPAMI*, 36(1):58–72, 2014. 7

[39] A. Milan, K. Schindler, and S. Roth. Multi-target tracking by discrete-continuous energy minimization. *TPAMI*, 38(10):2054–2068, 2016. 1

[40] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR*, 2011. 2

[41] S. H. Rezatofighi, A. Milan, Z. Zhang, Q. Shi, A. R. Dick, and I. D. Reid. Joint probabilistic data association revisited. In *ICCV*, 2015. 7

[42] A. Sadeghian, A. Alahi, and S. Savarese. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. In *ICCV*, 2017. 2, 7, 8

[43] R. Sanchez-Matilla, F. Poiesi, and A. Cavallaro. Multi-target tracking with strong and weak detections. In *ECCVW*, 2016. 7

[44] X. Shi, H. Ling, J. Xing, and W. Hu. Multi-target Tracking by Rank-1 Tensor Approximation. In *CVPR*, 2013. 1

[45] S. Tang, M. Andriluka, B. Andres, and B. Schiele. Multiple people tracking by lifted multicut and person re-identification. In *CVPR*, 2017. 7, 8

[46] Z. Wu, A. Thangali, S. Sclaroff, and M. Betke. Coupling detection and data association for multiple object tracking. In *CVPR*, 2012. 1

[47] Y. Xiang, A. Alahi, and S. Savarese. Learning to track: Online multi-object tracking by decision making. In *ICCV*, 2015. 2, 6, 7

[48] X. Yan, X. Wu, I. A. Kakadiaris, and S. K. Shah. To track or to detect? an ensemble framework for optimal selection. In *ECCV*, 2012. 2

[49] B. Yang and R. Nevatia. Online learned discriminative part-based appearance models for multi-human tracking. In *ECCV*, 2012. 2

[50] J. H. Yoon, M.-H. Yang, J. Lim, and K.-J. Yoon. Bayesian multi-object tracking using motion context from multiple objects. In *WACV*, 2015. 2

[51] A. R. Zamir, A. Dehghan, and M. Shah. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In *ECCV*, 2012. 2

[52] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *CVPR*, 2008. 2

[53] L. Zhao, X. Li, J. Wang, and Y. Zhuang. Deeply-learned part-aligned representations for person re-identification. In *ICCV*, 2017. 6