POI: Multiple Object Tracking with High Performance Detection and Appearance Feature

Fengwei Yu$^{1,3(\boxtimes)}$, Wenbo Li^{2,3}, Quanquan Li^3, Yu Liu^3,
Xiaohua Shi^1, and Junjie Yan^3

^{1} Beihang University, Beijing, China
forwil@buaa.edu.cn
^{2} University at Albany, SUNY, Albany, USA
^{3} Sensetime Group Limited, Beijing, China

Abstract. Detection and learning based appearance feature play the central role in data association based multiple object tracking (MOT), but most recent MOT works usually ignore them and only focus on the hand-crafted feature and association algorithms. In this paper, we explore the high-performance detection and deep learning based appearance feature, and show that they lead to significantly better MOT results in both online and offline setting. We make our detection and appearance feature publicly available (https://drive.google.com/open?id=0B5ACiy41McAHMjczS2p0dFg3emM). In the following part, we first summarize the detection and appearance feature, and then introduce our tracker named Person of Interest (POI), which has both online and offline version (We use POI to denote our online tracker and KDNT to denote our offline tracker in submission.).

1 Detection

In data association based MOT, the tracking performance is heavily affected by the detection results. We implement our detector based on Faster R-CNN [14]. In our implementation, the CNN model is fine-tuned from the VGG-16 on ImageNet. The additional training data includes ETHZ pedestrian dataset [4], Caltech pedestrian dataset [2] and the self-collected surveillance dataset (365653 boxes in 47556 frames). We adopt the multi-scale training strategy by randomly sampling a pyramid scale for each time. However, we only use a single scale and a single model during test. Moreover, we also use skip pooling [1] and multi-region [5] strategies to combine features at different scales and levels.

In considering the definition of MOTA in MOT16 [12], the sum of false negatives (FN) and false positives (FP) poses a large impact on the value of MOTA. In Table 1, we show that our detection optimization strategies lead to the significant decrease in the sum of FP and FN[1].

---

[1] We use detection score threshold 0.3 for Faster R-CNN and -1 for DPMv5, labeling the ID of detection box with incremental integer, and evaluate FP and FN with MOT16 devkit.

Table 1. Detection performance evaluation (on MOT16 train set)

Strategies	FP	FN	FP + FN
DPMv5	28839	62353	91192
Faster R-CNN baseline	5384	47343	52727
Faster R-CNN + skip pooling	5410	46399	51809
Faster R-CNN + multi-region	4476	46738	51214
Faster R-CNN + both	8722	37865	46587

2 Appearance Feature

The distance between appearance features is used for computing the affinity value in data association. The affinity value based on the ideal appearance feature should be large for persons of the same identity, and be small for persons of different identities. In our implementation, we extract the appearance feature using a network which is similar to GoogLeNet [15]. The input size of our network is 96×96, and the kernel size of *pool*5 layer is 3×3 instead of 7×7. The output layer is a fully connected layer which outputs the 128 dimensional feature. In the tracking phase, patches are first cropped according to the detection responses, and then resized to 96×96 for feature extraction. The cosine distance is used for measuring the appearance affinity.

For training, we collect a dataset which contains nearly 119 K patches from 19835 identities. Such a dataset consists of multiple person re-id datasets, including PRW [18], Market-1501 [18], VIPeR [13] and CUHK03 [8]. We use the softmax and triplet loss jointly during training. The softmax loss guarantees the discriminative ability of the appearance feature, while the triplet loss ensures the cosine distance of the appearance features of the same identity to be small.

3 Online Tracker

We implement a simple online tracker, which uses Kalman filter [6] for motion prediction and Kuhn-Munkres algorithm [7] for data association. The overall tracking procedure is described in Algorithm 1.

In the following, we introduce the affinity matrix construction, data association method, threshold value setting and tracking quality metric.

Affinity Matrix Construction. To construct an affinity matrix for the Kuhn-Munkres algorithm, we calculate the affinity between tracklets and detections. We combine motion, shape and appearance affinity as the final affinity. Specifically, the appearance affinity is calculated based on the appearance feature described in Sect. 2. Details of the affinity calculation are given below:

Algorithm 1. Overall Procedure of the Online Tracker

Input: A new frame at the t-th timestep, the detection set D^t, and the tracklet set T^{t-1}

Output: The new tracklet set T^t

1: Calculate the affinity matrix $A^{t-1} = Affinity(T^{t-1}, D^t)$
2: Divide T^{t-1} into high tracking quality set T_{high}^{t-1} and low quality set T_{low}^{t-1} with threshold τ_t
3: Use Kuhn-Munkres algorithm to find the optimal matching between $(T_{high}^{t-1}, T_{low}^{t-1})$ and D^t based on A^{t-1}
4: Use threshold τ_a to decide whether association success or not
5: Obtain association-success set $T_{success_i}^{t-1}$ with matched detection set $D_{success_i}^t$, association-fail tracklet set T_{fail}^{t-1} and unmatched detection set D_{fail}^t
6: Use Kalman filter and feature aggregation to generate new tracklet subset T_1^t based on association-success set: $T_1^t = Average(T_{success_i}^{t-1}, D_{success_i}^t)$.
7: Use Kalman filter to predict or remove the association-fail tracklets with missing tracklets threshold τ_m: $T_2^t = Predict\_Or\_Remove(T_{fail}^{t-1}, \tau_m)$.
8: Initialize the unmatched detections as the new tracklets: $T_3^t = Initialize(D_{fail}^t)$.
9: Merge the tracklet subsets to generate new candidate tracklet set : $T_{candidate}^t = T_1^t \cup T_2^t \cup T_3^t$.
10: Remove out of image border candidate tracklet set to generate new tracklet set: $T^t = Filter(T_{candidate}^t)$

$$aff_{app}(trk_i, det_j) = cosine(feat_{trk_i}, feat_{det_j}) \tag{1}$$

$$aff_{mot}(trk_i, det_j) = e^{-w_1 * ((\frac{X_{trk_i} - X_{det_j}}{W_{det_j}})^2 + (\frac{Y_{trk_i} - Y_{det_j}}{H_{det_j}})^2)} \tag{2}$$

$$aff_{shp}(trk_i, det_j) = e^{-w_2 * (\frac{|H_{trk_i} - H_{det_j}|}{H_{trk_i} + H_{det_j}} + \frac{|W_{trk_i} - W_{det_j}|}{W_{trk_i} + W_{det_j}})} \tag{3}$$

$$affinity(trk_i, det_j) = aff_{app}(trk_i, det_j) * aff_{mot}(trk_i, det_j) * aff_{shp}(trk_i, det_j) \tag{4}$$

aff_{app}, aff_{mot} and aff_{shp} indicate appearance, motion and shape affinity between the detection and tracklet, respectively. We combine these affinities with weights w_1 and w_2 as the final affinity.

Data Association. The tracklets and new detections are associated using the Kuhn-Munkres algorithm. Since the Kuhn-Munkres algorithm attempts to yield the global optimal result, it may fail when some detections are missing. To this end, we use a two-stage matching strategy, which divides T^{t-1} into high tracking quality set T_{high}^{t-1} and low quality set T_{low}^{t-1}. The matching is first performed between T_{high}^{t-1} and D, and then performed between $(T_{high}^{t-1} - T_{success}^{t-1}) \cup T_{low}^{t-1}$ and $D - D_{success}$.

Threshold Value Setting. On line 2 of Algorithm 1, we introduce τ_t to divide T^{t-1} into high and low tracking quality set. The strategy is intuitive: we mark a tracklet with *high* flag whose tracking quality is higher than τ_t, other tracklets will be mark as *low*. On line 4, we use τ_a to mark the association to be success or fail based on the affinity value. On line 7, we use τ_m as a threshold to drop a tracklet which is lost for more than τ_m frames.

Tracking Quality Metric. Tracking quality is designed to measure whether a object is tracking well or not. We use following formula to define tracking quality:

$$Quality(tracklet_i) = \frac{\sum_{k \in couples(tracklet_i)} \mathit{affinity}_k}{length(tracklet_i)}(1 - e^{-w_3 * \sqrt{length(tracklet_i)}})$$
(5)

where $couples(tracklet_i)$, with the form $\{trk_x, det_y\}$, is a set that contains every success association couple in history.

4   Offline Tracker

Our offine tracker an improved version of H²T [16] while based on K-Dense Neighbors [11]. It is more robust and efficient than H²T in handling the complex tracking scenarios. The overall procedure of the tracker is described in Algorithm 2.

Algorithm 2. Overall Procedure of the Offline Tracker

Input: A tracking video and the detections in all frames
Output: The tracking results (trajectories of targets)
 1: Divide the tracking video into multiple disjoint segments in the temporal domain
 2: Use the dense neighbors (DN) search² to associate the detection responses into short tracklets
        in each segment
 3: **while** The number of segments is greater than one **do**
 4:        Merge several nearby segments into a longer segment
 5:        Use the DN search in each longer segment to associate existing tracklets into longer tracklets
 6: **end while**

We make the following improvements over H²T [16].

Appearance Representation. To construct the affinity matrix for the dense neighbors (DN) search, we need to calculate three affinities, *i.e*, appearance, motion, and smoothness affinity. Among these three affinities, the appearance affinity is the most important one and we use the CNN based feature described in Sect. 2, instead of the hand-crafted feature in [16].

Big Target. A scenario that H²T [16] does not work well is the mixture of small and big targets. The reason is that the motion and smoothness affinities are unreliable for the big targets. Such unreliability is caused by the unsteady detection responses of the big targets. We introduce two thresholds, τ_s and τ_r, regarding the object scale to deal with this challenge, *i.e*, τ_s for preventing associating detection responses from very different scale, and τ_r for determining whether to reduce the weights of motion and smoothness affinity. Specifically,

---

² The DN search is performed on an affinity matrix which encodes the similarity between two tracklets. Please refer to [3,10,11,16] for details about DN search and its advantages over the GMCP [9,17] as a data association method.

if the ratio of the detection response scale and the target scale is less than τ_s, such a detection response will not be associated with the target. If the ratio of the detection response height and the image height is greater than τ_r, such a detection response will not be associated with the target. Both τ_s and τ_r are set as 0.5.

Algorithm Efficiency. H^2T is slow in handling the long tracking sequence where there exist plenty of targets. Among the steps in the algorithm, the step of DN search is the most time-consuming. To be more specific, the larger an affinity matrix, the longer time it will take to perform the DN search. Thus, we abandon the high-order information [16] when constructing the affinity matrix, which significantly reduces the matrix dimensions and improves the algorithm efficiency.

5   Evaluation

Our online and offline tracker are not learning based algorithm. We only tuning detection score threshold on train set and apply it to its similar scene from test set. For evaluation and submission, 0.1 is set for MOT16-03 and MOT16-04 due to high precision of detection result (03 and 04 are both surveillance scene, which is quite easy while our detector have been trained by self-collected surveillance dataset), and 0.3 is set for other sequences.

For both online[3] and offline tracker, we compare our detector with the official detector, and compare our feature with default CNN feature. The comparison results on MOT16 [12] train set are listed in Tables 2 and 3, respectively. Note that our detector leads to much better results in MT, ML, FP and FN, and our feature helps reduce both IDS and FM.

Table 2. Online tracker result on the train set

Det. and Feat.	MT	ML	FP	FN	IDS	FM	MOTA	MOTP
DPMv5 + Our Feat.	7.54 %	52.42 %	6197	70952	784	2697	29.4	77.2
Our Det. + GoogLeNet Feat.	31.72 %	16.25 %	**3207**	35472	1541	2235	63.6	**82.6**
Our Det. and Feat.	**37.33 %**	**14.70 %**	3497	**34241**	**716**	**1973**	**65.2**	82.4

Table 3. Offline tracker result on the train set

Det. and Feat.	MT	ML	FP	FN	IDS	FM	MOTA	MOTP
DPMv5 + Our Feat.	10.64 %	52.80 %	27238	63443	1540	1853	16.5	77.4
Our Det. + GoogLeNet Feat.	13.93 %	60.93 %	**1258**	58213	1350	2196	44.9	**85.0**
Our Det. and Feat.	**37.52 %**	**17.60 %**	2762	**33327**	**462**	**717**	**66.9**	83.3

---

[3] we use 0.5 for w_1, 1.5 for w_2, 1.2 for w_3, 0.5 for τ_t, 0.4 for τ_a and 100 frames for τ_m.

Table 4. Comparison to the state-of-the-art methods on MOT16 rank list

Tracker	MT	ML	FP	FN	IDS	FM	MOTA	MOTP
KFILDAwSDP (online)	26.9 %	21.6 %	23266	56394	1977	2954	55.2	77.2
MCMOT-HDM (offline)	31.5 %	24.2 %	9855	57257	1394	1318	62.4	78.3
Our online tracker	33.99 %	20.82 %	**5061**	55914	**805**	3093	66.1	**79.5**
Our offline tracker	**40.97**%	**18.97**%	11479	**45605**	933	**1093**	**68.2**	79.4

6   ECCV 2016 Challenge Results

Our ECCV 2016 Challenge results are listed in Table 4. Obviously, both our online and offline trackers outperform the state-of-the-art approaches by a large margin. Note that our offline tracker achieves the best performance in FN. However, its performance in FP is moderate, due to the interpolation module.

7   Conclusion

In this submission, we take many efforts to get high performance detection and deep learning based appearance feature. We show that they lead to the state-of-the-art multiple object tracking results, even with very simple online tracker. One observation is that with high performance detection and appearance feature, the state-of-the-art offline tracker does not have expected advantages over the much simpler online one. This observation is not reported in many current MOT papers, which often use detections that are not good enough. We make our detections and deep learning based re-ID features on MOT2016 publicly available, and hope that they can help more sophisticated trackers to get better performance.

References

1. Bell, S., Zitnick, C.L., Bala, K., Girshick, R.B.: Inside-outside net: detecting objects in context with skip pooling and recurrent neural networks. CoRR (2015)
2. Dollár, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: a benchmark. In: CVPR (2009)
3. Du, D., Qi, H., Li, W., Wen, L., Huang, Q., Lyu, S.: Online deformable object tracking based on structure-aware hyper-graph. TIP (2016)
4. Ess, A., Leibe, B., Schindler, K., Gool, L.J.V.: A mobile vision system for robust multi-person tracking. In: CVPR (2008)
5. Gidaris, S., Komodakis, N.: Object detection via a multi-region and semantic segmentation-aware CNN model. In: ICCV (2015)
6. Kalman, R.E.: A new approach to linear filtering and prediction problems. J. Basic Eng. **82**, 35–45 (1960)
7. Kuhn, H.W.: The hungarian method for the assignment problem. Naval Res. Logistics Q. **2**, 83–97 (1955)
8. Li, W., Zhao, R., Xiao, T., Wang, X.: Deepreid: deep filter pairing neural network for person re-identification. In: CVPR (2014)

9. Li, W., Wen, L., Chuah, M.C., Lyu, S.: Category-blind human action recognition: a practical recognition system. In: ICCV (2015)
10. Li, W., Wen, L., Chuah, M.C., Zhang, Y., Lei, Z., Li, S.Z.: Online visual tracking using temporally coherent part cluster. In: WACV (2015)
11. Liu, H., Yang, X., Latecki, L.J., Yan, S.: Dense neighborhoods on affinity graph. IJCV **98**(1), 65–82 (2012)
12. Milan, A., Leal-Taixé, L., Reid, I.D., Roth, S., Schindler, K.: MOT16: a benchmark for multi-object tracking. CoRR (2016)
13. Prosser, B., Zheng, W., Gong, S., Xiang, T.: Person re-identification by support vector ranking. In: BMVC (2010)
14. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: NIPS (2015)
15. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S.E., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: CVPR (2015)
16. Wen, L., Li, W., Yan, J., Lei, Z., Yi, D., Li, S.Z.: Multiple target tracking based on undirected hierarchical relation hypergraph. In: CVPR (2014)
17. Roshan Zamir, A., Dehghan, A., Shah, M.: GMCP-tracker: global multi-object tracking using generalized minimum clique graphs. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part II. LNCS, vol. 7573, pp. 343–356. Springer, Heidelberg (2012)
18. Zheng, L., Zhang, H., Sun, S., Chandraker, M., Tian, Q.: Person re-identification in the wild. CoRR (2016)