Online Multi-object Tracking Using Single Object Tracker and Markov Clustering

Jiao Zhu, Shanshan Zhang(✉), and Jian Yang

PCA Lab, Key Lab of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, and Jiangsu Key Lab of Image and Video Understanding for Social Security, School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China
{jiaozhu,shanshan.zhang,csjyang}@njust.edu.cn

Abstract. In this paper, we address the challenging problem of online Multi-Object Tracking (MOT). We find for those targets that are occluded or too small, the detectors usually fail to locate them. But an SOT tracker always provides a prediction for each target in the next frame. Therefore, we propose to use Single Object Tracking (SOT) predictions as complementary to detections. Also, we solve the data association problem via a new clustering method based on the Markov Clustering Algorithm (MCL). We first build a graph based on the targets, detections and SOT predictions, and then separate different identities by clustering. Experimental results on the MOT17 benchmark shows that our proposed method outperforms previous state-of-the-art methods w.r.t. MOTA and also reduces the number of false negatives and fragments significantly.

Keywords: Multi-object Tracking · Single Object Tracking · Markov Clustering

1 Introduction

Multiple object tracking (MOT) in videos serves as a fundamental and important task for many vision applications, such as video surveillance and autonomous driving. The purpose of this task is to locate multiple objects in each frame and obtain the trajectory for each identity. Most recently proposed approaches for MOT adopt the tracking-by-detection framework, which formulates the tracking problem as data association and solves it by linking detections frame by frame [3,25,30–32,38]. According to different requirements of application systems, MOT can be handled in either offline or online mode. The offline mode makes full use of all frames across the entire sequences to generate trajectories; in contrast, the online mode only has access to previous frames and the current frame. In this paper, we focus on the online mode, which is more challenging and is required by most online systems.

As we all know, the association algorithm is critical for the multiple object tracking task. For online MOT, a conventional way is to perform matching among

Y. Zhao et al. (Eds.): ICIG 2019, LNCS 11903, pp. 555–567, 2019.
https://doi.org/10.1007/978-3-030-34113-8_46

(a) (b)

Fig. 1. Illustration of missing detections and recovered MOT results from our proposed method. Two major factors that cause missing detections are (a) occlusion and (b) small scale. Top row: detections. Bottom row: our MOT results. The two sequences are from MOT17-04 and MOT17-02 sets, respectively.

detections in neighboring frames. Concurrent methods have made great efforts on learning effective feature representations for matching [32,37]. A big disadvantage of those methods is that they rely on the provided detections, which are sometimes noisy and they are not able to recover from missing detections. We show two examples in the upper row of Fig. 1, where the missing detections are caused by occlusion and small scale. Several works [43–45] have a great progress for pedestrian detection, but it is quite time consuming for these detectors to make efficient work for recognizing and understanding video sequences with complex scenes.

In order to overcome the above problems, some methods propose to use single object tracking (SOT) predictions as compensation [7,47]. The SOT method predicts the location of an identity in the next frame given the location in the current frame. Previous methods rely too much on the SOT predictions, resulting in frequent drifts towards other identities in complex scenes. As of now, it still remains an open question how to integrate the detections and SOT predictions, which are independent to each other.

Therefore, in this paper we investigate how to integrate pre-generated detections and single object tracking predictions in a unified framework. We propose a new graph clustering algorithm to locally cluster three groups of bounding boxes on two neighboring frames: the MOT targets on frame $t-1$; the SOT predictions on frame t; and the detections on frame t. After clustering, the target location of each identity on frame t will be estimated by all the bounding boxes belonging to its cluster. The MOT results on frame t are then used as targets when processing frame t and $t+1$.

In summary, our contributions are as follows:

– In order to compensate for noisy and missing detections, we propose to consider SOT predictions and integrate two sources of bounding boxes in a more balanced manner for online MOT.

- We propose a new graph clustering procedure using the Markov Cluster Algorithm (MCL) algorithm to locally cluster MOT targets, SOT predictions and detections into different identities according to similarities represented by deep features.
- From the experimental results on the challenging MOT17 benchmark, we demonstrate that our method achieves state-of-the-art results among online methods. As shown in Fig. 1b, our approach is able to recover missing detections so as to obtain more complete trajectories.

2 Related Work

Multi-object Tracking Using SOT Tracking. Some previous works [7,40,47] have attempted to use single object tracking approaches to solve the MOT problem. Zhu *et al.* [47] design a cost-sensitive tracking loss based on ECO [9] tracker and propose Dual Matching Attention Networks with spatial and temporal attention mechanisms. It relies too much on the single object tracking predictions without making full use of detections. Chu *et al.* [7] use CNN-based single object tracker with spatial-temporal attention mechanism to handle the drift caused by occlusion and interaction among targets, but it does not consider how to deal with missing targets. Different from previous works, we integrate detections and single object tracking predictions in a more balanced way to estimate the targets' final locations. The single object tracker runs independently to track targets even when they are occluded.

Multi-object Tracking by Data Association. Data association is important for the MOT task. Most online processing methods [3,13,38] adopt Hungarian Algorithm [26] to match detections and targets. Wojke *et al.* [38] propose a simple online and real-time tracking method with a deep association metric, but it depends too much on the quality of detections and features based on the appearance and position. On the other hand, offline methods consider MOT task as a global optimization problem by using the multi-cut model [30–32] or network flow [10,36,42]. For detection based graph models, it is effective to fix noisy detections, but is hard to find the global optimal solution. In this paper, we borrow the idea of graph clustering from offline MOT, but reduce the scale of the graph from global to local by a large margin. In this way, our method is able to fix noisy detections but it makes the optimization problem much easier to solve.

3 Online MOT Framework

The framework of the proposed online MOT algorithm is shown in Fig. 2. First, an SOT tracker is used to make prediction for each target at frame t (see Sect. 3.1). All bounding boxes of targets, SOT predictions and detections are cropped into image patches for further processing. Second, an affinity graph

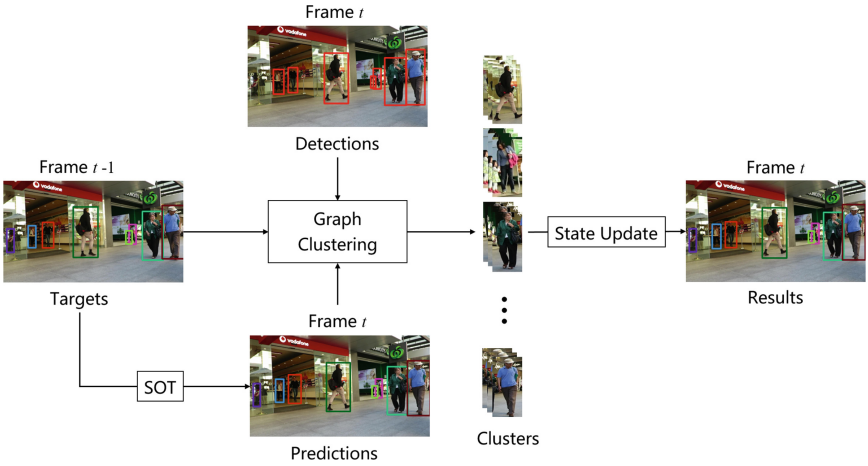


Fig. 2. The proposed online MOT framework. The graph clustering is performed on top of the targets, predictions and detections. Each cluster consists of a group of image patches from the same identity. After clustering, we update the location of each target in frame t by taking into account both SOT predictions and detections.

model is built based on the whole set of image patches. After that, we utilize a new clustering procedure to partition all image patches into groups, one for an identity (see Sect. 3.2). Finally, we update location of each target at frame t according to the predictions and detections inside the cluster (see Sect. 3.3).

In the following, we will describe each component of our framework in more detail.

3.1 SOT Algorithm

For a tracking-by-detection framework, the MOT performance very much rely on the quality of detections. When the detector fails to find a tiny or occluded object, the trajectory becomes broken and a wrong ID switch may happen in this frame. Fortunately, the SOT method can be used to recover missing detections.

In this paper, we choose the Discriminative Correlation Filter Tracker with Channel and Spatial Reliability (DCF-CSR) [23] for tracking each single object. The spatial reliability map adapts the correlation filters to support to the part of the object during tracking. This strategy enlarges the search region when the target happens to be occluded. The channel reliability scores which reflect channel-wise quality of the learned filters, are used for weighting the per-channel filter responses. The DCF-CSR tracker obtains state-of-the-art results on several standard object tracking benchmarks, including OTB100 [39], VOT2015 [18] and VOT2016 [17]. It also runs in real-time on a single CPU as it uses computationally efficient features, i.e. HoG [8] and Colornames [33].

Given a set of D-channel features $F = \{f_1, ..., f_D\}$ and correlation filters $H = \{h_1, ..., h_D\}$, the location corresponding to the maximum value in the correlation

response indicates the new position of the target. Additionally, the DCF-CSR tracker introduces channel reliability weights $W = \{w_1, ..., w_D\}$ that considered as scaling factors based on the discriminative power of each feature channel.

$$\tilde{Y} = \sum_{l=1}^{D} f_l \star h_l \cdot w_l. \tag{1}$$

Here, the symbol $\star$ represents circular correlation computation between features f_l and filters h_l. The optimal correlation filters H are estimated by minimizing the following cost function:

$$\varepsilon(H) = \sum_{l=1}^{D} \|f_l \star h_l - Y\|^2 + \lambda \|h_l\|^2, \tag{2}$$

where the variable Y is the desired output, which is a 2-D Gaussian function centred at the target location, and λ is a regularization parameter that controls overfitting.

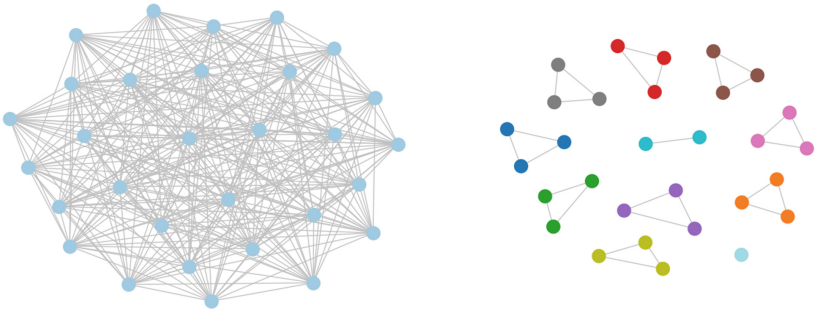
3.2 Graph Clustering

We solve the data association problem via a graph clustering method. Different from previous works, our graph is constructed based on two adjacent frames with local information. Since the number of clusters is unknown, we use the Markov Cluster Algorithm (MCL) [34] to partition the graph into multiple sub-graphs.

Graph Definition. For every two adjacent frames $t-1$ and t, we first define a finite set V, which consists of a series of bounding boxes: the targets at frame $t-1$, the predictions by SOT tracker and the provided detections at frame t. Another finite set E is composed of edges. Each element $e \in E$ represents an edge between two nodes $v, w \in V$. Every edge $e \in E$ has a cost, represented by the similarity $c \in (0, 1)$ computed based on deep feature of two nodes. A weighted and undirected graph $G = (V, E)$ shown in Fig. 3a is then defined with the following two constraints:

- For $v, w \in V$, if both of them come from the same category among the targets, SOT predictions and detections, they should not be connected, the edge $\{v, w\} \notin E$.
- For $v, w \in V$, if they are too far way in either the spatial domain or the feature domain, they should not be connected, the edge $\{v, w\} \notin E$.

Clustering. Given an affinity graph, we apply our proposed clustering algorithm to partition it into clusters, each of which consists of bounding boxes of one single identity. We show an illustration in Fig. 3b.



(a) (b)

Fig. 3. Illustration of our graph clustering method. (a) In the graph, each node indicates one bounding box; each edge represents affinity between a pair of nodes. We measure similarity with CNNs features. (b) Nodes are grouped into different clusters, each of which consists of bounding boxes of one identity.

To partition the graph thoroughly, we develop a new graph clustering method by running the Markov Cluster Algorithm (MCL) for multiple rounds. The MCL algorithm finds cluster structure in a graph by a mathematical bootstrapping procedure. It simulates random walks through the graph by alternation of two operators called expansion and inflation. Expansion coincides with taking the power of the graph matrix using the normal matrix product (i.e. matrix squaring), and allows flow to connect different regions of the graph. Inflation corresponds with taking the Hadamard power of the graph matrix, and changes the probabilities associated with the collection of random walks. Shortening the expansion parameter and increasing the inflation parameter are able to improve the granularity or tightness of clusters.

In our method, we run the MCL algorithm for multiple times to reach reasonable numbers of predictions and detections in each cluster. The detail of our graph clustering is illustrated in Algorithm 1. First, The MCL process is applied on the whole graph and obtains coarse clusters where sometimes one node is contained in multiple clusters or alone in a cluster. Then, we adapt the inflation parameter step by step and perform a loop graph clustering (Algorithm 2) on overlapping clusters where multiple targets are connected. After that, we adopt the loop graph clustering again on a sub-graph consisting of all incomplete clusters so as to make sure each detection node connect to its target. Here, incomplete clusters indicate those ones missing SOT predictions or detections.

3.3 State Update

After clustering, we classify all clusters into four different types according to the number of prediction and detection boxes in each cluster. We show an illustration in Fig. 4.

Algorithm 1. Graph Clustering

Input: Affinity Graph $G = (V, E)$.
Output: Confirmed Clusters Set M.

$M \leftarrow \varnothing$, incomplete clusters set $D \leftarrow \varnothing$;
Apply MCL algorithm to cluster on Graph G to discover clusters set C;
for $c \in C$ **do**
  $n1$, $n2$, $n3 \leftarrow$ the numbers about target, prediction and detection in c;
  if $n1 = 0$ **or** $n2 = 0$ **or** $n3 \geq 0$ **then**
    $D = D \cup c$;
  else if $n1 = 1$ **and** $n2 = 1$ **and** $n3 \geq 1$ **then**
    $M = M \cup c$;
  else
    Get a sub-graph g by a overlapping cluster c;
    Loop graph clustering (Algorithm 2) on g to get clusters set S;
    for $s \in S$ **do**
      $k1$, $k2$, $k3 \leftarrow$ the numbers about target, prediction and detection in s;
      if $k1 = 1$ **and** $k2 = 1$ **and** $k3 \geq 1$ **then**
        $M = M \cup s$;
      else
        $D = D \cup s$;
      end if
    end for
  end if
end for
Get a sub-graph g by D;
Loop graph clustering (Algorithm 2) on g to get clusters set S;
for $s \in S$ **do**
  $M = M \cup s$;
end for

For each type of cluster, we design a corresponding state update strategy, and explain different strategies in the following:

(a) The state is estimated by merging the SOT prediction and detection(s).
(b) The state is first estimated by Kalman filter prediction and then refined by merging the prediction and detection(s).
(c) The state is estimated by the SOT prediction.
(d) The target is seen as out of view, so we remove it from the MOT list.

4 Experiments

Dataset. We evaluate our proposed online MOT method on the MOT17 benchmark dataset [24]. The dataset consists of 7 videos for train and 7 videos for test. Each video sequence is provided with 3 sets of detections, i.e. DPM [12], Faster-RCNN [27] and SDP [41].

Algorithm 2. Loop Graph Clustering

Input: Sub-graph $G = (V, E)$, Initial inflation parameter r, Increment Δr.
Output: Clusters Set C.

```
NOT_OK ← true;
while NOT_OK do
    Cluster on Graph G with inflation parameter r to discover clusters set C;
    for c ∈ C do
        n1, n2, n3 ← the numbers about target, prediction and detection in c;
        if n1 = 0 or n2 = 0 or n3 = 0 then
            NOT_OK ← false;
        else if n1 = 1 and n2 = 1 and n3 >= 1 then
            NOT_OK ← false;
        else
            NOT_OK ← true;
            break;
        end if
        r ← r + Δr;
    end for
end while
```

Fig. 4. Illustration of four cluster types. We classify clusters into four types only for target including (a) One target, one prediction and one or more detections; (b) One target and one or more detections; (c) One target and one prediction; (d) One target. The circle, triangle, square indicate targets, SOT predictions and detections, respectively.

Evaluation Metrics. We adopt the evaluation metrics defined in [2,19,22,24,28]: Multiple Object Tracking Accuracy (MOTA) [2], Multiple Object Tracking Precision (MOTP) [2], ID F1 score (IDF1) [28], the ratio of Mostly Tracked targets (MT), the ratio of Mostly Lost targets (ML), the number of False Positives (FP), the number of False Negatives (FN), the number of Identity Switches (IDS) [22] and the number of fragments (Frag). In these metrics, we mainly force on MOTA which can intuitively measure the performance of tracker. As illustrated in Eq. (3), MOTA combines three error sources: false positives (FP), missed targets (FN) and identity switches (IDS).

$$MOTA = 1 - \frac{\sum_t (FN_t + FP_t + IDS_t)}{\sum_t GT_t} \tag{3}$$

Implementation Details. We call the DCF-CSR tracker by OpenCV tracking API which contains implementations of many single object tracking algorithms. To reduce drifts, the tracker always serves the final updated location as template.

Fig. 5. Tracking examples from the MOT-CVPR19 challenge. The top tracklet of ID 6 and the bottom tracklet of ID 401 in our results are from CVPR19-01 set and CVPR19-03 set respectively.

Table 1. Tracking performance on the test set of the MOT17 benchmark dataset.

Mode	Method	MOTA↑	MOTP↑	IDF1↑	MT↑	ML↓	FP↓	FN↓	IDS↓	Frag↓
Online	GMPHD_N1Tr [1]	42.1	**77.7**	33.9	11.9%	42.7%	**18,214**	297,646	10,698	10,864
	EAMTT [29]	42.6	76.0	41.8	12.7%	42.7%	30,711	288,474	4,488	5,720
	FPSN [20]	44.9	76.6	48.4	16.5%	**35.8%**	33,757	269,952	7,136	14,491
	PHD_GSDL [14]	48.0	77.2	49.6	17.1%	35.6%	23,199	265,954	3,998	8,886
	AM_ADM [21]	48.1	76.7	52.1	13.4%	39.7%	25,061	265,495	2,214	5,027
	DMAN [47]	48.2	75.9	**55.7**	19.3%	38.3%	26,218	263,608	**2,194**	5,378
	Ours	**48.4**	76.3	45.5	**19.4%**	35.9%	33,525	**255,091**	2,531	**4,944**
Offline	MHT_bLSTM [16]	47.5	**77.5**	**51.9**	18.2%	41.7%	25,981	268,042	**2,069**	**3,124**
	IOU [4]	45.5	75.9	39.4	15.7%	40.5%	**19,993**	281,643	5,988	7,404
	EDMT [6]	**50.0**	77.3	51.3	**21.6%**	**36.3%**	32,279	**247,297**	2,264	3,260

If the target is only tracked by single object tracker over a period of time $t_{max} = 30$, it will be seen as out of view and be removed from MOT list. We employ a pre-trained CNN model [37] trained on a large-scale person re-id dataset [46] to extract deep feature with 128 dimensionality. The affinity graph is based on the cosine distance of pair-wise deep feature with thresholds about the feature domain and the space domain: $\tau_f = 0.2$ and $\tau_s = 9.4877$.

4.1 Results on the MOT Benchmark Datasets

We evaluate our proposed method on the test sets of the MOT17 benchmark and compare it with the state-of-the-art MOT trackers in Table 1. The symbol "↑" means that higher is better and the symbol "↓" means that lower is better.

Compared to other online methods, our MOT method achieves the best performance in terms of MOTA, MT, FN and Frag metrics. Especially, our method

Table 2. Comparison performance with different SOT trackers on the train set of the MOT17 benchmark dataset.

SOT	MOTA↑	MOTP↑	IDF1↑	MT↑	ML↓	FP↓	FN↓	IDS↓	Frag↓
MOSSE [5]	27.5	81.2	20.7	1.9%	81.5%	**3,374**	236,840	3,178	6,643
KCF [15]	49.8	**81.5**	48.8	12.2%	64.8%	11,457	155,130	2,601	2,436
UDT [35]	50.4	81.1	**50.4**	**13.1%**	**63.7%**	14,201	151,360	**1,683**	**2,062**
DCF-CSR [23]	**50.6**	**81.5**	50.0	13.0%	63.9%	14,214	**150,660**	1,719	2,085

Table 3. Contributions of SOT and clustering.

Method	MOTA↑	FP↓	FN↓	IDS↓	(FP + FN + IDS) ↑
Baseline	47.7	7,802	165,041	3,354	176,197
+ clustering	48.2	**7,463**	165,050	2,044	174,557
+ SOT	49.1	21,523	**147,783**	2,140	171,446
+ SOT, clustering	**50.6**	14,214	150,660	**1,719**	**166,593**

has far achievements than other online methods in terms of FN and Frag. The scores of FN and Frag precisely explain that our method can fix the problems caused by missing detections. Besides, the performance of our method in MOTA is also near with state-of-art offline methods performance. Also we show some instances of our results in Fig. 5. Those tracklets are from MOT CVPR 2019 challenge [11], which was released not long ago and is hiding other results now. Therefore, we are not able to compare our method with other ones, but we can visualize tracking results, that will be helpful to find success and failure cases.

4.2 Ablation Study

SOT Algorithm Selection. In terms of performance of single object tracking, we compare MOT results with different state-of-the-art SOT methods on the train sets of the MOT17 benchmark dataset. We use MOSSE [5], KCF [15], DCF-CSR [23] and UDT [35] respectively to get different results. Among these SOT methods, MOSSE, KCF and DCF-CSR are all correlation filter trackers based on different hand-crafted features, while UDT tracker is an unsupervised correlation filter tracking method with deep features. As illustrated in Table 2, DCF-CSR and UDT have pretty performance in terms of all metrics. But considering about running speed and the value of MOTA, we finally choose DCF-CSR as the single object tracker in our MOT method.

Impact of SOT and Clustering. We set up different experiments to demonstrate the contributions of SOT algorithm and graph clustering. First, we associate only targets and detections by building an assignment problem that can be solved by the Hungarian Algorithm [26]. Second, we consider the data association as a local optimization by adopting the MCL algorithm. Last, we add single

object tracking module to previous experiments. As illustrated in Table 3, clustering works better than assignment, and the method with single object tracking performs better than one without single object tracking. In general, SOT and clustering modules have positive effects on the performance of MOT.

5 Conclusions

In this paper, we introduce a unified online multi-object tracking framework which integrates single object tracking predictions and pre-generated detections, and applies graph clustering to solve local optimization. For single object tracking, we use DCF-CSR tracker to track each target location. For graph clustering, we take the MCL algorithm repeatedly to reach reasonable cluster results. In the end, we evaluate our proposed method on the MOT benchmark dataset and obtain better performance than other state-of-the-art trackers.

Acknowledgments. The authors would like to thank the editor and the anonymous reviewers for their critical and constructive comments and suggestions. This work was supported by the National Natural Science Foundation of China (Grant No. 61702262, U1713208), Program for Changjiang Scholars, Funds for International Cooperation and Exchange of the National Natural Science Foundation of China (Grant No. 61861136011), Natural Science Foundation of Jiangsu Province, China (Grant No. BK20181299), CCF-Tencent Open Fund (RAGR20180113), "the Fundamental Research Funds for the Central Universities" No.30918011322, and Young Elite Scientists Sponsorship Program by CAST (2018QNRC001).

References

1. Baisa, N.L., Wallace, A.: Development of a N-type GM-PHD filter for multiple target, multiple type visual tracking. J. Vis. Commun. Image Represent. **59**, 257–271 (2019)
2. Bernardin, K., Stiefelhagen, R.: Evaluating multiple object tracking performance: the clear mot metrics. J. Image Video Process **2008**, 1–10 (2008)
3. Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B.: Simple online and realtime tracking. In: ICIP (2016)
4. Bochinski, E., Eiselein, V., Sikora, T.: High-speed tracking-by-detection without using image information. In: AVSS (2017)
5. Bolme, D.S., Beveridge, J.R., Draper, B.A., Lui, Y.M.: Visual object tracking using adaptive correlation filters. In: CVPR (2010)
6. Chen, J., Sheng, H., Zhang, Y., Xiong, Z.: Enhancing detection model for multiple hypothesis tracking. In: CVPR Workshop (2017)
7. Chu, Q., Ouyang, W., Li, H., Wang, X., Liu, B., Yu, N.: Online multi-object tracking using CNN-based single object tracker with spatial-temporal attention mechanism. In: ICCV (2017)
8. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005)

9. Danelljan, M., Bhat, G., Shahbaz Khan, F., Felsberg, M.: Eco: efficient convolution operators for tracking. In: CVPR (2017)
10. Dehghan, A., Tian, Y., Torr, P.H., Shah, M.: Target identity-aware network flow for online multiple target tracking. In: CVPR (2015)
11. Dendorfer, P., et al.: CVPR19 tracking and detection challenge: how crowded can it get? arXiv preprint arXiv:1906.04567 (2019)
12. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Trans. Pattern Anal. Mach. Intell. **32**(9), 1627–1645 (2010)
13. Feng, W., Hu, Z., Wu, W., Yan, J., Ouyang, W.: Multi-object tracking with multiple cues and switcher-aware classification. arXiv preprint arXiv:1901.06129 (2019)
14. Fu, Z., Feng, P., Angelini, F., Chambers, J., Naqvi, S.M.: Particle PHD filter based multiple human tracking using online group-structured dictionary learning. IEEE Access **6**, 14764–14778 (2018)
15. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. IEEE Trans. Pattern Anal. Mach. Intell. **37**(3), 583–596 (2015)
16. Kim, C., Li, F., Rehg, J.M.: Multi-object tracking with neural gating using bilinear LSTM. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11212, pp. 208–224. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01237-3_13
17. Kristan, M., et al.: The visual object tracking vot2016 challenge results. In: ICCV Workshop (2016)
18. Kristan, M., et al.: The visual object tracking vot2015 challenge results. In: ICCV Workshop (2015)
19. Leal-Taixé, L., Milan, A., Reid, I., Roth, S., Schindler, K.: Motchallenge 2015: towards a benchmark for multi-target tracking. arXiv preprint arXiv:1504.01942 (2015)
20. Lee, S., Kim, E.: Multiple object tracking via feature pyramid siamese networks. IEEE Access **7**, 8181–8194 (2019)
21. Lee, S.H., Kim, M.Y., Bae, S.H.: Learning discriminative appearance models for online multi-object tracking with appearance discriminability measures. IEEE Access **6**, 67316–67328 (2018)
22. Li, Y., Huang, C., Nevatia, R.: Learning to associate: hybridboosted multi-target tracker for crowded scene. In: CVPR (2009)
23. Lukežič, A., Vojíř, T., Čehovin Zajc, L., Matas, J., Kristan, M.: Discriminative correlation filter with channel and spatial reliability. In: CVPR (2017)
24. Milan, A., Leal-Taixé, L., Reid, I., Roth, S., Schindler, K.: Mot16: a benchmark for multi-object tracking. arXiv preprint arXiv:1603.00831 (2016)
25. Milan, A., Schindler, K., Roth, S.: Multi-target tracking by discrete-continuous energy minimization. IEEE Trans. Pattern Anal. Mach. Intell. **38**(10), 2054–2068 (2016)
26. Munkres, J.: Algorithms for the assignment and transportation problems. SIAM **5**(1), 32–38 (1957)
27. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: NIPS (2015)
28. Ristani, E., Solera, F., Zou, R., Cucchiara, R., Tomasi, C.: Performance measures and a data set for multi-target, multi-camera tracking. In: Hua, G., Jégou, H. (eds.) ECCV 2016. LNCS, vol. 9914, pp. 17–35. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48881-3_2

29. Sanchez-Matilla, R., Poiesi, F., Cavallaro, A.: Online multi-target tracking with strong and weak detections. In: Hua, G., Jégou, H. (eds.) ECCV 2016. LNCS, vol. 9914, pp. 84–99. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48881-3_7
30. Tang, S., Andres, B., Andriluka, M., Schiele, B.: Subgraph decomposition for multi-target tracking. In: CVPR (2015)
31. Tang, S., Andres, B., Andriluka, M., Schiele, B.: Multi-person tracking by multicut and deep matching. In: Hua, G., Jégou, H. (eds.) ECCV 2016. LNCS, vol. 9914, pp. 100–111. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48881-3_8
32. Tang, S., Andriluka, M., Andres, B., Schiele, B.: Multiple people tracking by lifted multicut and person re-identification. In: CVPR (2017)
33. Van De Weijer, J., Schmid, C., Verbeek, J., Larlus, D.: Learning color names for real-world applications. IEEE Trans. Image Process. **18**(7), 1512–1523 (2009)
34. Van Dongen, S.M.: Graph clustering by flow simulation. Ph.D. thesis (2000)
35. Wang, N., Song, Y., Ma, C., Zhou, W., Liu, W., Li, H.: Unsupervised deep tracking. In: CVPR (2019)
36. Wang, X., Türetken, E., Fleuret, F., Fua, P.: Tracking interacting objects using intertwined flows. IEEE Trans. Pattern Anal. Mach. Intell. **38**(11), 2312–2326 (2016)
37. Wojke, N., Bewley, A.: Deep cosine metric learning for person re-identification. In: WACV (2018)
38. Wojke, N., Bewley, A., Paulus, D.: Simple online and realtime tracking with a deep association metric. In: ICIP (2017)
39. Wu, Y., Lim, J., Yang, M.H.: Object tracking benchmark. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1834–1848 (2015)
40. Xiang, Y., Alahi, A., Savarese, S.: Learning to track: online multi-object tracking by decision making. In: ICCV (2015)
41. Yang, F., Choi, W., Lin, Y.: Exploit all the layers: fast and accurate CNN object detector with scale dependent pooling and cascaded rejection classifiers. In: CVPR (2016)
42. Zhang, L., Li, Y., Nevatia, R.: Global data association for multi-object tracking using network flows. In: CVPR (2008)
43. Zhang, S., Benenson, R., Omran, M., Hosang, J., Schiele, B.: How far are we from solving pedestrian detection? In: CVPR (2016)
44. Zhang, S., Benenson, R., Omran, M., Hosang, J., Schiele, B.: Towards reaching human performance in pedestrian detection. IEEE Trans. Pattern Anal. Mach. Intell. **40**(4), 973–986 (2017)
45. Zhang, S., Yang, J., Schiele, B.: Occluded pedestrian detection through guided attention in CNNs. In: CVPR (2018)
46. Zheng, L., et al.: MARS: a video benchmark for large-scale person re-identification. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9910, pp. 868–884. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46466-4_52
47. Zhu, J., Yang, H., Liu, N., Kim, M., Zhang, W., Yang, M.-H.: Online multi-object tracking with dual matching attention networks. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11209, pp. 379–396. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01228-1_23