Challenges on Large Scale Surveillance Video Analysis

Weitao Feng[1], Deyi Ji[2], Yiru Wang[1], Shuorong Chang[1], Hansheng Ren[3], Weihao Gan[4]
[1]Beihang University, China
[2]Shanghai Jiao Tong University, China
[3]University of Chinese Academy of Sciences, China
[4]University of Southern California, USA

Abstract

Large scale surveillance video analysis is one of the most important components in the future artificial intelligent city. It is a very challenging but practical system, consists of multiple functionalities such as object detection, tracking, identification and behavior analysis. In this paper, we try to address three tasks hosted in NVIDIA AI City Challenge contest. First, a system that transforming the image coordinate to world coordinate has been proposed, which is useful to estimate the vehicle speed on the road. Second, anomalies like car crash event and stalled vehicles can be found by the proposed anomaly detector framework . Third, multiple camera vehicle re-identification problem has been investigated and a matching algorithm is explained. All these tasks are based on our proposed online single camera multiple object tracking (MOT) system, which has been evaluated on the widely used MOT16 challenge benchmark. We show that it achieves the best performance compared to the state-of-the-art methods. Besides of MOT, we evaluate the proposed vehicle re-identification model on VeRi-776 dataset and it outperforms all other methods with a large margin.

1. Introduction

Nowadays, with the development of computer vision technologies, large scale surveillance video analysis for AI city draws more and more attention in the real world applications. It is a very powerful but challenging system that can identify the person-of-interest, locate the suspicious vehicle and detect the anomaly event. In order to achieve those tasks, several important components need to be properly addressed, such as multiple object tracking, object re-identification.

The multiple object tracking (MOT) technique predicts locations of multiple objects and maintains their identities to yield their individual motion trajectories throughout a video sequence. Existing MOT solutions can be categorized into two classes: 1) global optimization methods and 2) online methods. Global optimization methods [4, 8, 15, 29] minimize the total energy cost from all target trajectories. They examine all detection results of each frame and link fragmented trajectories due to occlusion. To build a more accurate energy affinity measure, a “tracklet” is defined across multiple consecutive frames and exploited to extract the spatial and temporal features of the target. The major drawback of global optimization is that it is not suitable for real-time applications.

In contrast, In contrast, online MOT methods are designed for real-time applications. Online MOT solutions have been studied in [1, 3, 19, 21]. The trajectory of each target is constructed frame by frame fashion, where the location and identity of one target are determined by the information of the current frame without accessing future frames. The most challenging task in online MOT is to find an appropriate target matching model that correctly connects detection results of the current frame to tracks obtained from previous frames.

For vehicle re-identification problem, the goal is to identify all the images of the same vehicle from a large gallery dataset. Such task is particularly useful when the car license plate is occluded. Vehicle reID methods can be used in these scenarios to effectively locate vehicles of interest from surveillance datasets. Compared with the problem of person reID, vehicle reID is a recently proposed research topic with several challenging factors: (1) the number of the different vehicle makes and models is small and the appearance information can be very similar, while face or clothing information can be a very distinguishable feature for person reID; (2) usually vehicle moves along a fixed direction without rotation, which results in invisible key information for re-identification, while the human behavior is much more social so that the key features like face show time to time.

In this work, we focus on solving some real world problems presented in NVIDIA AI City Challenge like multiple object tracking, speed estimation and vehicle re-identification. The rest of this paper is organized as follows. Section 2 briefly reviews the existing work of multiple object tracking and vehicle re-identification. Our proposed methods for each task are explained in Section 3. Experimental results are shown in Section 4. Finally, we conclude the work and show the future direction in Section 5.

2. Related Work

Global optimization MOT. With the advancement of object detection techniques [6, 7], tracking-by-detection becomes popular for multiple objects tracking. In order to find the trajectory of each target from detection results in all frames, data association is an essential task. It is usually conducted in a discrete space using the linear programming or graph-based methods. Various optimization algorithms such as the network flow [16, 32], the continuous energy minimization [15] and the subgraph multi-cut [24] have been proposed. Several energy cues were introduced and optimized using the standard conjugate gradient method in [15]. In [5], each target trajectory is generated one by one in the optimization process from the best clique to the next.

Online methods. Several online MOT methods [1, 28] have been proposed recently to tackle with the practical real-time tracking applications. Under the ”online” requirement, the ID association problem is more challenging since there are occlusions and interactions among objects. The focus has been on developing an online matching model that has an accurate feature representation so as to associate the current target location with previously detected trajectory. The part-based feature tracking was exploited in [19] to handle partial occlusion. The recurrent neural networks (RNNs) were used in [14] and [17] to manage the spatial and temporal consistency of different targets.

Vehicle re-identification. This is a relatively new proposed research topic that has not received much attention. Recent works on it mainly concentrate on building retrieval pipelines and benchmarks. [11] built a high-quality multi-viewed vehicle reID dataset (VeRi-776) with 776 vehicle identities. Another large surveillance vehicle reID dataset (VehicleID) is proposed by [10], which contains more than 20,000 identities. And Coupled Clusters Loss (CCL) is proposed for performance evaluation on it. There are also some pioneering works [18, 26] on vehicle reID problems that achieve promising results.

3. Proposed Solutions

3.1. Multiple object detection and tracking

We track the vehicles in a tracking-by-detection manner and follow the online tracking protocal. We first build a detection model by which most vehicles can be detected, then we do detection during each frame, and associate bounding boxes between frames. Finally, we divide all detected bounding boxes into several sets, one denotes one ID.

3.1.1 Detection model

A DenseNet architecture is used in our detection model. We did not make any advanced optimization on the network architecture nor make any big difference about training methods. We just fine-tune the model using a large self-labelled vehicles dataset in addition to academical public datasets that currently exist.

3.1.2 Bounding box association

Consider two frames nearby, the part which consists of tracked targets and the other part of detected objects to be tracked in new frame form a bi-party graph. For each two objects i and j, where i is one of the tracked targets and j is a new detection, there is an edge weighted W_{ij} between them. Here we set W_{ij} as $1 - IoU(i, j)$, function $IoU(\cdot)$ calculates the Intersection of Union between a pair of bounding boxes. In this way, we can work out the best matching pairs using Minimum-cost-maximum-flow (MCMF) by setting edge capacity to 1 (unit flow). For better accuracy, we set $w_\theta = 0.7$ as a weight threshold to dismiss invalid edges and avoid bad matching pairs, i.e., only edges weighted smaller than w_θ will be considered.

After matching, the ones in tracked targets that does not match any detection for a few frames will be regarded as disappeared items. Correspondingly, detections without matching will be insert into tracked set as a new target. For each matching pair, box position will be updated according to the new detection box.

The association steps are described as algorithm 1.

3.2. Traffic flow analysis

We aim to calculate the speed of vehicles in videos by our MOT result. Naturally, we try to transform points from image coordinate system to world coordinate system. On the condition that both extrinsic and intrinsic camera parameters are unknown, we assume the road is a plane, then lane width and stripes length can be used to work out a plane-to-plane transform.

Consider the transform between a 3D point $M = [x, y, z]$ and its image projection $m = [u, v]$:

$$s\tilde{m} = A[R \;\; t]\tilde{M}$$

$$\tilde{m} = [u, v, 1]^T$$

$$\tilde{M} = [x, y, z, 1]^T$$

where s is an arbitrary scale factor, $[R \;\; t]$ is the rotation and translation matrix which relates the world coordinate system to the camera coordinate system, A is camera intrinsic matrix, which relates the camera coordinate system to the image coordinate system.

As $[R \;\; t]$ denote the conversion between world coordinate system and camera coordinate system, $[R \;\; t]\tilde{M}$ is point

Algorithm 1 MOT association

```
Input: Set of tracked targets at frame $t$ $\Gamma_t$ and detection set in frame $t+1$ $D_{t+1}$
Output: Set of tracked targets at frame $t+1$ $\Gamma_{t+1}$
procedure AssociateBBox($\Gamma_t, D_{t+1}$)
    Initialize weight matrix $W_{ij}$ with value INFINITY
    for each target $\gamma_t^i$ in $\Gamma_t$ do
        for each detection $d_{t+1}^j$ in $D_{t+1}$ do
            if $1 - IoU(\gamma_t^i, d_{t+1}^j) < w_\theta$ then
                Set $W_{ij}$ to $1 - IoU(\gamma_t^i, d_{t+1}^j)$
            end if
        end for
    end for
    Find best matchings of matrix $W_{ij}$ using MCMF
    Set $\Gamma_{t+1}$ to empty
    for each target $\gamma_t^i$ in $\Gamma_t$ do
        if $\gamma_t^i$ is matched then
            Update position of $\gamma_t^i$
            Add $\gamma_t^i$ to $\Gamma_{t+1}$
        else if $\gamma_t^i$ not been matched for T frames then
            Delete $\gamma_t^i$
        end if
    end for
    for each detection $d_{t+1}^j$ in $D_{t+1}$ do
        if $d_{t+1}^j$ is NOT matched then
            Add $d_{t+1}^j$ to $\Gamma_{t+1}$
        end if
    end for
    Return $\Gamma_{t+1}$
end procedure
```

M's coordinates in camera coordinate system. We denote $[R\ \ t]\tilde{M}$ as $[P_x, P_y, P_z]$. Then we set $s = P_z$.

Additionally, We set road plane as $X - Y$ plane in the world coordinate system. The relationship between a 2D point $N = [x, y]$ which is on the road and its image projection $n = [u, v]$ is as the following:

$$P_z \tilde{n} = H\tilde{N}$$

where $\tilde{n} = [u, v, 1]^T, \tilde{N} = [x, y, 1]^T$, $H = A[R\ \ t]$ is a 3×2 Matrix. For each plane, We can work out the above transform parameters through Gaussian Elimination using 4 reference point-pairs. Using bounding box information given by MOT, we can get all object's world locations. We also use some other technique to avoid large errors: divide the road into two planes on account of road's convexity, use multiple points' reconversion precision to supervise the selection of reference points. As the results from MOT are sometimes trembling, we apply Kalman Filter to make the trajectory smoother.

All candidate reference points (image coordinates) are detected by lane line detection algorithm which aims to find all stripe area with equal width, and then these points are classified to different class according to U.S. standard and rewritten as reference point-pairs.

For more details, after all vehicles' world locations are calculated, we assume that one's speed is stable during a short time window $[t-\epsilon, t+\epsilon]$ and output the average speed of the time window as one's speed at time t. Here we set ϵ as 1/6 s, i.e., 5 frames at 30 fps video.

3.3. Anomaly detection

Vehicle anomaly detection is a very practical task in surveillance video. The anomalies are defined as car crashes and stalled vehicles. In this section, we introduce the proposed framework on this task.

The system consists of four stages: vehicle detection and tracking, data cleaning, track merging and anomaly detection:

- Vehicle detection and tracking: we apply the proposed multiple object detection and tracking system for each video to get all the vehicle tracks. The detection area threshold is set to be 5x5 pixels.

- Data cleaning: due to the low video quality in the task, we observe some false positive detections in the previous step. Most of them locate in the texture background region outside of the road. If the system keeps receiving the information from those detections, the anomaly detector will be triggered. Therefore, we propose a way to find the road region, illustrated in Figure 1. All the non-static vehicle tracks have been recorded to generate a heat-map and the area of the road can be inferred by this heat-map. Then the detections out of the road will be removed.

- Track merging: in order to generate a full trajectory of a vehicle for the future anomaly detection, it is necessary to remove the fragment situation. We define a shallow neural network for this merging mission using vehicle reID feature, position, speed and size information. The reID appearance feature is a 256-dimension vector generated by our proposed method explained in Section 3.4.1. Tracks with high network score will be merged together.

- Anomaly detection: to detect the anomalous track, we measure the time duration of the vehicle existence. The anomalies are defined as car crashes and stalled vehicles, which usually last for a long time period in the video. Therefore, if the duration of a vehicle is much longer than the average duration of all the vehicles in the video, we claim it as an anomaly. This also shows the necessity of our data cleaning step to remove the false positive detections.

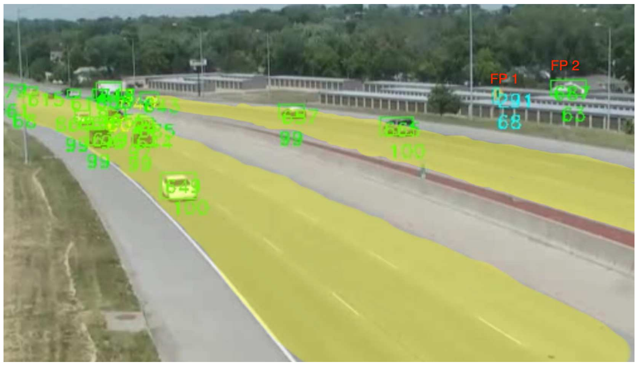

Figure 1. Road detection: all the non-static tracks are recorded to generate a heat-map (yellow), which indicates the road area. According to this information, the false positive detections (FP 1 and FP 2) out of the road can be removed.

3.4. Multi-camera vehicle re-identification

3.4.1 Appearance reID feature

The reID feature extraction branch is built upon on GoogLeNet [23] architecture, which is proposed for handling muti-scale objects simultaneously, as well as computational efficiency and practicality. The main component of GoogLeNet is the ”Inception Layer”, which convolves the input image on different scales, from the fine-grained level (reID) to the coarse one(5*5). Table 1 indicates the structure of our network. The output of our feature extraction branch is a 256-dimension feature vector, which not only includes the semantic feature for the object but also the spatial detailed information for object parts. We include the experimental results for this feature extraction branch to show the vehicle re-identification performance in Section 4.

Table 1. The proposed feature extraction network for vehicle reID

type	patch size/stride	output size	depth
convolution	3*3/1	192*192*32	1
convolution	3*3/2	96*96*32	1
convolution	3*3/1	96*96*64	1
max pool	2*2/2	48*48*64	0
inception(4a)		48*48*256	3
inception(4b)		24*24*384	3
inception(4d)		24*24*512	3
inception(4e)		12*12*768	3
inception(4f)		12*12*1024	3
inception(4g)		6*6*1536	3
ave pool	6*6/1	1*1*256	0
dropout(0.7)		1*1*256	0
linear		1*1*256	1
triplet loss		1*1*256	0

To train the network for re-identification task, the triplet loss is adopted in our work. The main idea of triplet loss is to minimize the distance between an anchor and a positive sample (same identity), and maximizes the distance between the anchor and a negative sample (different identity). A triplet unit consists of an anchor x_i with its corresponding positive sample x_i^p and negative sample x_i^n. The loss function is defined as:

$$L = \sum_{i=1}^{N} [\|f(x_i) - f(x_i^p)\|_2^2 + \alpha - \|f(x_i) - f(x_i^n)\|_2^2]_+$$

where $f(x)$ is the appearance feature extraction network, $[\cdot]_+ = max\{\cdot, 0\}$ and α is a parameter which defines the minimum margin between matched and mismatched pairs.

3.4.2 Spatio-temporal cue

Appearance feature is a very powerful cue to identify object instants, especially for pedestrian reID scenarios because of personalized decorations. However, it is may not be adequate enough to distinguish one vehicle from others when the vehicles are of the same model and the plate information is not accessible. Therefore, in order to refine the search results, we integrate the spatio-temporal cue into consideration.

Specifically, the spatio-temporal cue is a probability model of the relationship between location and time information when vehicle passing through different cameras. It measures how likely that a vehicle spends time duration τ from one specific camera to another. Similar to the idea in [26], we treat the vehicle transition interval between pairs of cameras as a random variable following the logarithmic normal distribution:

$$p(\tau \mid \mu, \sigma) = \ln\mathcal{N}(\tau; \mu, \sigma) = \frac{1}{\tau\sigma\sqrt{2\pi}}\exp\left[-\frac{(\ln\tau - \mu)^2}{2\sigma^2}\right]$$

where μ and σ are the parameters to be estimated for each camera pair. From each camera pair, we can collect all the time transition interval samples τ_n from the training set. By maximizing the log-likelihood function:

$$L(\tau \mid \mu, \sigma) = \prod_{n=1}^{N} \left(\frac{1}{\tau_n}\right) \mathcal{N}(\ln\tau_n; \mu, \sigma)$$

we have the estimated parameters as:

$$\hat{\mu} = \frac{\sum_{n=1}^{N} \ln\tau_i}{N}$$

$$\hat{\sigma}^2 = \frac{\sum_{n=1}^{N} (\ln\tau_i - \hat{\mu})^2}{N}$$

Therefore, besides of the calculated appearance distance D_a, we can measure the spatio-temporal similarity distance between two vehicles based on the above probability model as:

$$D_s = \frac{1}{1 + e^{\alpha p(\tau)}}$$

Finally, the similarity distance between two vehicles is defined as the weighted summation of those two cues: D_a and D_s.

3.4.3 Re-ranking

Object re-identification problem can be also treated as a retrieval process and therefore, we apply the re-ranking technique to improve the object search accuracy. The general idea is that after ranking the initial similarity distance matrix of the probe and gallery sets, the subsequent re-ranking is adopted within the k nearest neighbors of each probe. Then the final distance is computed as the combination of the initial distance and the re-ranked distance. Following the idea in [34], we also use the k-reciprocal encoding method to rerank the initial result and find the true match of the target. Through this process, we achieve around 6% improvement on mAP score during the evaluation. Refer to [34] for details as we use the same formulation and parameters.

3.4.4 Multi-camera multi-target (MCMT) tracking

In this part, we explain the overall tracking algorithm for the challenge Track 3 based on the components explained above including multiple object tracking, appearance and spatio-temporal feature extraction, re-ranking and ID matching process. Here are the detailed steps:

- Step 1: Generate all the individual tracks in all the single videos in the set with the multiple object tracking pipeline.

- Step 2: For each track in previous step, calculate the mean appearance feature from all the images and then generate the appearance distance matrix.

- Step 3: For each pair of tracks in Step 1, use the middle timestamp of each track to calculate the transition interval and evaluate the spatio-temporal distance matrix.

- Step 4: Combine the two distance matrices and perform the multi-camera multi-target matching following the Algorithm 2.

4. Experimental Results

In this section, we include the performance evaluation of two individual components: multiple object tracking and vehicle re-identification, on two public available datasets: MOTchallenge16 [13] and VeRi-776 [11], respectively. Also, we show our AICity challenge contest results here.

Algorithm 2 MTMC matching

```
    Input: Distance matrix $G$
    Output: Trajectory result list $R$
procedure MATCHTRACK($G$)
    Trajectory candidate list $R_c$
    for each row $g_i$ in $G$ do
        List $T = []$
        Sort $g_i$ to select 4 smallest ones from each video
        for each element $i$ in selected set do
            if $Rerank(i) < Thre$ then $T.append(i)$
            end if
        end for
        if $In4Loc(T)$ is True then $R_c.append(T)$
        end if
    end for
    Calculate in-group correlation for each row in $R_c$
    Sort $R_c$ and select top N trajectories into $R$
    Return $R$
end procedure
```

4.1. Multiple object tracking

MOTchallenge [13] benchmark is a widely used dataset for evaluating the performance of multiple object tracking. There are seven training and seven testing sequences. The target object is pedestrian and the detection results of all frames are available for reference. However, in order to provide a high detection rate, we adopted one of the most powerful detectors (private) from our own side. For evaluation, we follow the CLEAR metrics [22], including the multiple objects tracking accuracy (MOTA), the multiple objects tracking precision (MOTP), the false positives (FP), the false negatives (FN), most tracked (MT), most lost (ML), the identity switch error (IDs) and the total fragments of all the trajectories (Frag).

We compare the proposed multiple object tracking system with several the state-of-the-art methods on testing sequences of MOT2016. Similar to others, we use the private pedestrian detectors to get a high detection rate. Table 2 shows the performance result. Among all the published online methods, our proposed method (**FLOW4**) achieves the best performance in MOTA (67.7), MT (35.0%) and FN (49178). The MOTA score is also competitive among all the listed offline methods.

In the AICity Challenge contest, we use our vehicle detector and multiple object tracking system to generate the track for each vehicle in each video. All challenge tasks are based on this detection and tracking module.

Table 2. MOT16 tracking performance with private detector. In each mode (online/offline), the best performance is marked in bold text.

MOT16 - Test Set									
Tracker	Mode	MOTA	MT	ML	FP	FN	IDs	Frag	Hz
SORTwHPD16 [2]	**Online**	59.8	25.4%	22.7%	8698	63245	1423	**1835**	**59.5**
DeepSORT2 [27]	**Online**	61.4	32.8%	**18.2%**	12852	56668	**781**	2008	17.4
POI [31]	**Online**	66.1	34.0%	20.8%	**5061**	55914	805	3093	9.9
Ours	**Online**	**67.7**	**35.0%**	18.4%	8225	**49178**	1568	3153	24.7
MCMOTHDM [9]	Offline	62.4	31.5%	**24.2%**	9855	57257	1394	1318	**34.9**
KDNT [31]	Offline	68.2	41.0%	19.0%	11479	45605	933	1093	0.7
LMPp [25]	Offline	71.0	**46.9%**	21.9%	**7880**	44564	**434**	**587**	0.5
HTSJTUZTE	Offline	**71.3**	46.5%	19.5%	9238	**42521**	617	743	29

4.2. Vehicle re-identification

In order to train our appearance model, we collect the data from multiple datasets, including VeRi-776 [11], VehicleID [10], BoxCars21k [20], CompCars [30] and some self-labelled datas. In total, the training set contains more than 300,000 images of around 40,000 identities. For testing, we use the test set of VeRi-776 to evaluate our vehicle reID model. VeRi-776 [11] dataset is a large-scale benchmark dataset for vehicle Re-Id in the real-world urban surveillance scenario. It contains over 50,000 images of 776 vehicles captured by 20 cameras from different locations.

We follow the same evaluation metrics in VeRi-776, including mean average precision (mAP) and cumulative match curve (CMC). For each identity, one image is random selected from all the gallery images to generate the gallery set, while the probe set remains unchanged. The random selection procedure was repeated for 100 times to obtain an average CMC result.

In Table 3, we compare our method with different components, to several the state-of-the-art models, including BOW-CN [33], KEPLER [12], PROVID [11] and OIF [26]. In baseline method, we only use the appearance feature without re-ranking. We can see that our baseline already outperforms other listed methods. Our full version is consisted of appearance feature, spatio-temporal cue and reranking, which achieves 71.2 mAP score. We have shown that our proposed vehicle re-identification method improves the performance with 40% gain in mAP from OIF [26].

Besides of the standard evaluation setup in VeRi-776 dataset, we also use the training set to mimic a similar experimental environment compared to the challenge contest. In the new setup, the appearance feature of each image can be represented as the feature of one track from a video camera. The proposed MTMC tracking system groups images together to generate multiple trajectories. Therefore, we evaluate the performance and achieve around 0.6 tracking detection rate in VeRi-776 training set.

Table 3. VeRi-776 performance evaluation

Method	mAP	CMC1	CMC5
BOW-CN [33]	12.2	-	-
PROVID [11]	27.8	-	-
KEPLER [12]	33.5	48.2	64.3
OIF [26]	51.4	68.3	89.7
baseline	61.3	86.1	94.2
baseline + re-ranking	67.4	87.6	93.1
baseline + ST	68.1	88.2	**95.1**
full version	**71.2**	**89.3**	93.8

Table 4. Performance evaluation of challenge contest

	Track #1		Track #3	
Rank	ID	S1	ID	S3
1	48	1.0000	48	0.7106
2	79	0.9162	37	0.2861
3	78	0.8892	79	0.0785
4	24	0.8813	18	0.0074

4.3. AICity challenge contest

Here we (team ID **79**) report our challenge contest performance of the two tracks: traffic flow analysis and multi-camera vehicle detection and re-identification.

In track #1, our global score is 0.9162, which ranks number 2 in the overall evaluation. In track #3, we rank number 3 among all the teams. It is amazing that team48 performances so well in both track #1 and #3, and we will keep working on these tasks to improve our methods. Please check the following Table 4 for more details.

5. Conclusion and Future Work

In this paper, we introduce several very challenging but practical tasks in large scale surveillance video analysis hosted in NVIDIA AICity Challenge and explain the proposed methods to approach them. In this contest, one of the most fundamental components is multiple vehicle detection and tracking. We first propose a powerful online detection and tracking system in single camera as our starting point. Then, two main tasks: traffic flow analysis and multi-camera vehicle re-identification have been addressed properly. This is a very good opportunity for us to understand the difficulty of the real-world problems. In the future, we believe we will keep working on the related key problems, such as multiple object tracking and vehicle re-identification, to improve the large scale surveillance video analysis.

References

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

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

[3] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Online multiperson tracking-by-detection from a single, uncalibrated camera. *IEEE transactions on pattern analysis and machine intelligence*, 33(9):1820–1833, 2011.

[4] W. Brendel, M. Amer, and S. Todorovic. Multiobject tracking as maximum weight independent set. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1273–1280. IEEE, 2011.

[5] A. Dehghan, S. Modiri Assari, and M. Shah. Gmmcp tracker: Globally optimal generalized maximum multi clique problem for multiple object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4091–4099, 2015.

[6] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010.

[7] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[8] C.-H. Kuo and R. Nevatia. How does person identity recognition help multi-person tracking? In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1217–1224. IEEE, 2011.

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

[10] H. Liu, Y. Tian, Y. Yang, L. Pang, and T. Huang. Deep relative distance learning: Tell the difference between similar vehicles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2167–2175, 2016.

[11] X. Liu, W. Liu, T. Mei, and H. Ma. A deep learning-based approach to progressive vehicle re-identification for urban surveillance. In *European Conference on Computer Vision*, pages 869–884. Springer, 2016.

[12] N. Martinel, C. Micheloni, and G. L. Foresti. Kernelized saliency-based person re-identification through multiple metric learning. *IEEE Transactions on Image Processing*, 24(12):5645–5658, 2015.

[13] 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.

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

[15] A. Milan, S. Roth, and K. Schindler. Continuous energy minimization for multitarget tracking. *IEEE transactions on pattern analysis and machine intelligence*, 36(1):58–72, 2014.

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

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

[18] Y. Shen, T. Xiao, H. Li, S. Yi, and X. Wang. Learning deep neural networks for vehicle re-id with visual-spatio-temporal path proposals. *arXiv preprint arXiv:1708.03918*, 2017.

[19] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah. Part-based multiple-person tracking with partial occlusion handling. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1815–1821. IEEE, 2012.

[20] J. Sochor, A. Herout, and J. Havel. Boxcars: 3d boxes as cnn input for improved fine-grained vehicle recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3006–3015, 2016.

[21] X. Song, J. Cui, H. Zha, and H. Zhao. Vision-based multiple interacting targets tracking via on-line supervised learning. In *European Conference on Computer Vision*, pages 642–655. Springer, 2008.

[22] R. Stiefelhagen, K. Bernardin, R. Bowers, J. Garofolo, D. Mostefa, and P. Soundararajan. The clear 2006 evaluation. In *International Evaluation Workshop on Classification of Events, Activities and Relationships*, pages 1–44. Springer, 2006.

[23] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

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

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

[26] Z. Wang, L. Tang, X. Liu, Z. Yao, S. Yi, J. Shao, J. Yan, S. Wang, H. Li, and X. Wang. Orientation invariant feature embedding and spatial temporal regularization for vehicle re-identification. In *Proceedings of the IEEE Conference on*

Computer Vision and Pattern Recognition, pages 379–387, 2017.

[27] N. Wojke, A. Bewley, and D. Paulus. Simple online and realtime tracking with a deep association metric. *arXiv preprint arXiv:1703.07402*, 2017.

[28] S. C. Wong, V. Stamatescu, A. Gatt, D. Kearney, I. Lee, and M. D. McDonnell. Track everything: Limiting prior knowledge in online multi-object recognition. *IEEE Transactions on Image Processing*, 2017.

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

[30] L. Yang, P. Luo, C. Change Loy, and X. Tang. A large-scale car dataset for fine-grained categorization and verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3973–3981, 2015.

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

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

[33] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian. Scalable person re-identification: A benchmark. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1116–1124, 2015.

[34] Z. Zhong, L. Zheng, D. Cao, and S. Li. Re-ranking person re-identification with k-reciprocal encoding. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 3652–3661. IEEE, 2017.