Received June 11, 2018, accepted July 15, 2018, date of publication July 23, 2018, date of current version August 28, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2858853

Online Multi-Object Tracking via Combining Discriminative Correlation Filters With Making Decision

CHENGLONG WU [1,2,3], HAO SUN[1,3], HONGQI WANG[1,3], KUN FU[1,2,3,4], GUANGLUAN XU[1,3], WENKAI ZHANG[1,3], AND XIAN SUN[1,3]

[1]Institute of Electronics, Chinese Academy of Sciences, Beijing 100190, China
[2]School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 100190, China
[3]Key Laboratory of Technology in Geo-Spatial Information Processing and Application System, Institute of Electronics, Chinese Academy of Sciences, Beijing 100190, China
[4]Institute of Electronics, Chinese Academy of Sciences, Suzhou 215000, China

Corresponding author: Xian Sun (sunxian@mail.ie.ac.cn)

This work was supported by the National Natural Science Foundation of China under Grant 41501485.

ABSTRACT Multiple-object tracking (MOT) has received an increasing attention due to the rapid development of autonomous driving. However, the MOT problem is still challenging mainly due to the occlusion and scale variation. Motivated by the fact that the discriminative correlation filters-based (DCFB) tracking algorithms can tackle these problems and significantly improve the accuracy of single object tracking, how to exploit the DCFB tracking algorithms for MOT is worthy studying. Moreover, the corrupted training samples due to the occlusion make DCFB tracking methods to update the appearance model of target uncorrected and result in tracking drift. In this paper, we exploit Markov decision process to integrate the DCFB tracking method into our MOT framework and address the update problem of the appearance model in DCFB tracking method. Moreover, in order to overcome the challenges of occlusion and scale variation, to prevent target drift during tracking, we use two DCFB trackers with different update frequencies and a novel update strategy to predict the location of targets. The part-based method is used to extract robust features to tackling the challenges of occlusion and scale change. In order to verify the efficiency of our algorithm, experiments are performed based on KITTI tracking benchmark. The results demonstrate that our method achieves state-of-the-art performance and outperforms the state-of-the-art algorithms in road scenarios.

INDEX TERMS Multiple object tracking, Markov decision process, discriminative correlation filters, part-based method.

I. INTRODUCTION

Object tracking is an important subproblem of computer vision due to its widespread applications, such as security protections, visual analysis, human-computer interaction, autonomous driving, etc. It encompasses subfields called single object tracking (SOT) and multiple object tracking (MOT). The first one is to track a single target while the latter aims for tracking multiple targets. Specifically, for MOT, the locations of the interesting objectives are estimated during each frame of videos while the location of target is given in the first frame of a video in SOT. MOT is increasingly attractive due to the emergence of automatic driving. However, there are many challenges required to be tackle in order to implement MOT in practice, such as frequent occlusion, scale variation, and other disturbing factors [2], [3]. Thus, in this paper, we focus on MOT.

In the last decade, a great progress has been made for improving the tracking performance of SOT in terms of robustness and accuracy, especially when the algorithm based on discriminative correlation filters (DCF) and deep learning was proposed. The tracking algorithm based on DCF learns a discriminative appearance model in the first frame of a video, and then locates the target in the next frame and updates the appearance model according to the predicted location. Since the pioneering study of Bolme *et al.* [4], DCF-based (DCFB) tracking algorithm has drawn great attention due to its good performance and impressive tracking speed. A large number of investigations have been done to utilize the human extraction feature to learn and update the discriminative appearance model for tracking objects [8]–[14]. Meanwhile, deep neural networks have shown their strong representation learning ability in many applications, such as image classification [5],

recognition [6], semantic segmentation [7], etc. An intuitive thought is to combine deep neural networks with the algorithm based on DCF. In this case, deep neural networks can be used to extract robust feature while the algorithm based on DCF is exploited to predict the location of target [15]–[19]. It was shown that the performance and robustness can be significantly improved by this combination compared with the algorithm based on DCF with the human extraction feature. The DCFB tracking method has greatly improved the state-of-the-art accuracy of SOT.

Although the DCFB tracking method has made significant progress in the area of SOT, few investigations have focused MOT by using the DCFB tracking method. Motivated by the progress made for SOT, it is interesting and valuable to study MOT via combining DCFB tracking algorithm with deep neural networks.

Unfortunately, The tracking algorithm based on DCF for SOT cannot be directly applied for MOT due to the following challenges.

1) The performance of the tracking algorithm based on DCF for SOT depends on the effectiveness of the appearance model of target. The efficiency of the appearance model relies on the quality of training samples. However, the training samples are collected and labeled online which are decided by the predicted results of a single object tracker. In this case, the corrupted training samples can be collected due to the occlusion, scale variation and other distractions. Then, the appearance model is updated with those corrupted samples and results in gradually drifts to the distracter. Considering more frequent occlusion and scale variation in MOT scenarios especially in autonomous driving scenarios, this situation is much worse.
2) The updating frequency of the appearance model for the target is also a contradictory problem. Specifically, the tracker obtained by using the algorithm based on DCF cannot well fit the appearance change of the target when the appearance model is conservatively updated, and the tracking drift can be caused due to the corrupted training samples if the appearance model is aggressively updated.
3) The tracking algorithm based on DCF cannot efficiently tackle occlusion and the scale variant. However, the occlusion and scale variant are frequent in autonomous driving scenarios.

In this paper, we devote to tackling the above-mentioned challenges. Firstly, we construct the lifetime of a target as a MDP and integrate the DCFB SOT algorithm into our MOT framework. The appearance and disappearance of target, tracked and lost of the target, updating and non-updating of the appearance model are considered as states in MDP. In this way, the appearance model of target is not updated when the training sample is corrupted. Secondly, two DCBF trackers with different update frequencies of the appearance model are exploited to solve the contradictory problem of update frequency. One DCFB tracker updates the appearance model conservatively to maintain the appearance model of the target and to prevent drift to the corrupted training samples mainly due to occlusion, and the other one updates aggressively to well fit the appearance change of target. Finally, in order to address occlusion and scale variant, we partition a object into four sub-blocks, namely, the upper half, lower half, left half and right half part. Then the normalized cross correlation(NCC) values and correlation responses of these four sub-blocks are calculated with the corresponding templates. They are identified as features and can be used to learn the decision-making policy. These features make our approach robust to occlusion and scale variant.

The main contributions of our work are summarized as follows:

1) MDP is exploited to integrate the DCFB tracking method into our MOT framework. The appearance and disappearance of target, tracked and lost of target, updating and non-updating of the appearance model are all considered as states in MDP. In this way, the lifetime of a target is modeled as a MDP. Thus, we can use multiple MDPs to tackle the MOT problem well. Moreover, a pipeline is designed to show how to use the correlation response of the DCFB tracker for organizing multiple MDPs and addressing the conflict of different MDPs.
2) A policy is learned to decide whether the appearance model is updated or not in the tracking state. Two DCFB trackers with different updating frequencies of the appearance model are applied to overcome the contradictory problem caused by the updating frequency. The above two measures are simultaneously used and make our MOT approach more robust to the corrupted training samples mainly due to occlusion and improve the performance of our method. Moreover, a novel update strategy of the appearance model for the target is proposed to make our MOT method more robust to scale variant.
3) The part-based method is used to tackle the frequent occlusion and scale variant in the autonomous driving scenarios.
4) Experiments on KITTI tracking benchmark demonstrate that our method can significantly improve the performances compared with some state-of-the-art MOT algorithms. Ablation study shows the efficiency of our setting.

The rest of this paper is organized as follows. Section II presents the related work for the DCFB SOT algorithms and MOT algorithms. The most common means to handle occlusion and scale variant are also discussed in this section. The idea of our work is presented in Section III. Section IV introduces the famous tracking benchmark in autonomous driving: KITTI tracking benchmark, and presents experiment results in this benchmark. Finally, this paper is concluded in Section V.

II. RELATED WORK

A. SOT BASED ON DISCRIMINATIVE CORRELATION FILTERS

Due to the pioneering study of Bolme *et al.* [4], the tracking algorithm based on DCF has received great attention in the object tracking domain. It was shown that this algorithm can achieve great performance in terms of both the tracking accuracy and the tracking speed. These two metrics are important for trackers in the practical applications. Based on the work [4], a large number of investigations have been done to improve the accuracy and robustness of the tracking algorithm based on DCF. For example, Danelljan *et al.* [10]–[13] have done many works to improve the performance of the DCFB tracking method. In [10], a scale correlation filter was proposed to hold the scale change of targets in video. A novel color feature was extracted for robust tracking in [11]. They also proposed a spatial regularization to alleviate the unwanted boundary effects in [12]. A novel method was proposed to alleviate the negative effect of corrupted training samples in [14]. Circular structure and kernelized trick were exploited for the DCFB SOT in [8] and [9]. Multiple channel features were also utilized in these work. Since deep convolutional neural networks (CNNs) are promising in many computer vision applications [5]–[7], many works have been explored to combine deep CNNs with the DCFB method for online tracking. Ma *et al.* [16] exploited the low-level and high-level feature maps of deep CNNs to construct multiple DCFB trackers for SOT. Feature maps with different resolutions of deep CNNs were map into the continuous space so that the tracking algorithm based on DCF can well tackle multiple channel features with different resolutions and perform more accurate tracking [21], [22]. The works about the DCFB method with deep CNNs refer to these works in [17]–[19]. In our work, the tracking algorithm based on DCF proposed in [16] is adopted for SOT. Two base DCFB trackers with different update frequencies are used to address occlusions and other disturbance term.

B. MOT IN TRACKING-BY-DETECTION PARADIGM

In the past decade, since a great progress has been made in object detection [24]–[26], researches in MOT were mainly focused on tracking-by-detection paradigm. MOT algorithms in tracking-by-detection fashion detect objects in each frame of videos, and then use a data association algorithm to cooperatively detect the same target in different frames. Note that data association is the main challenge in MOT. The MOT algorithms in tracking-by-detection paradigm can be classified into two categories: the batch/ global method and the online method. Comparing with the online method, the batch method has a better performance since the history information and future information of target are utilized. A common approach for data association in the batch method is to find a network flow or graph matching [27]–[29] for minimizing the sum of pairwise association costs. Other than the pairwise association costs for data association which were learned as linear functions, pairwise association cost was learned in the end-to-end fashion via backpropagation in [31]. A novel hand-crafted feature named Aggregated Local Flow Descriptor (ALFD) has been introduced in [30]. Combining this hand-crafted feature with motion/appearance models, this batch MOT algorithm can obtain the state-of-the-art performance. Although the batch method has been well studied in many works, it is inappropriate for online MOT tasks. The reason is that it needs the future information of the target.

For the online method, intersection-over-union (IOU) overlap of detections in the previous and the subsequent frame have been used for MOT, and achieved 100K fps runing speed with the advanced performance in [32]. Pairwise association costs were constructed with different manners in [33] and [34]. The highly relative works to our work are [20] and [35]. Xiang *et al.* [35] have utilized MDP for MOT. We combine MDP with DCF for MOT. Whether the appearance mode is updated or not is integrated into the MDP. Different from the work in [20] that the CNN-based single object tracker was applied, the DCFB single object tracker is exploited for MOT in our work. Moreover, the data association algorithm, the treatment of occlusion and scale variant are different.

C. OCCLUSION AND SCALE VARIATION HANDLING IN OBJECT TRACKING

Occlusion and scale variation are the common problems in MOT especially in autonomous driving scenarios. The part-based idea is the most common approach for addressing occlusion in the object tracking [36]–[38]. The part-based method has been introduced into the DCFB method for SOT and well addressed the partial occlusion issue in [39]. For the scale variation, a scale correlation filter was proposed to choose the best scale of target in [11]. Moreover, the appearance model of the target was used to calculate the response of multiple scale region, and the scale which has the maximum response was chose as the best scale in [23]. In our work, the famous part-based method is used to handle occlusion and scale variation. Although the DCFB-based SOT tracker of our MOT algorithm [16] does not use any method to overcome the scale variation and occlusion, using the part-based method, two DCFB trackers with different update frequencies and a novel update strategy, our method outperforms all the existing methods under the road scenarios where scale change and occlusion are frequently happened.

D. DECISION MAKING IN OBJECT TRACKING

Since the attractive performance of reinforcement learning in Go and atari games [40]–[42], decision making has drawn an increasing attention. Object tracking has been extensively studied in the decision making fashion. The MDP-based method was proposed for SOT and MOT in [35] and [43]. The prioritized Q-learning algorithm was utilized to optimize the control parameters of SOT tracker in [43]. With the rise of deep reinforcement learning, Yun *et al.* [44] have studied SOT by using the deep reinforcement learning algorithm.

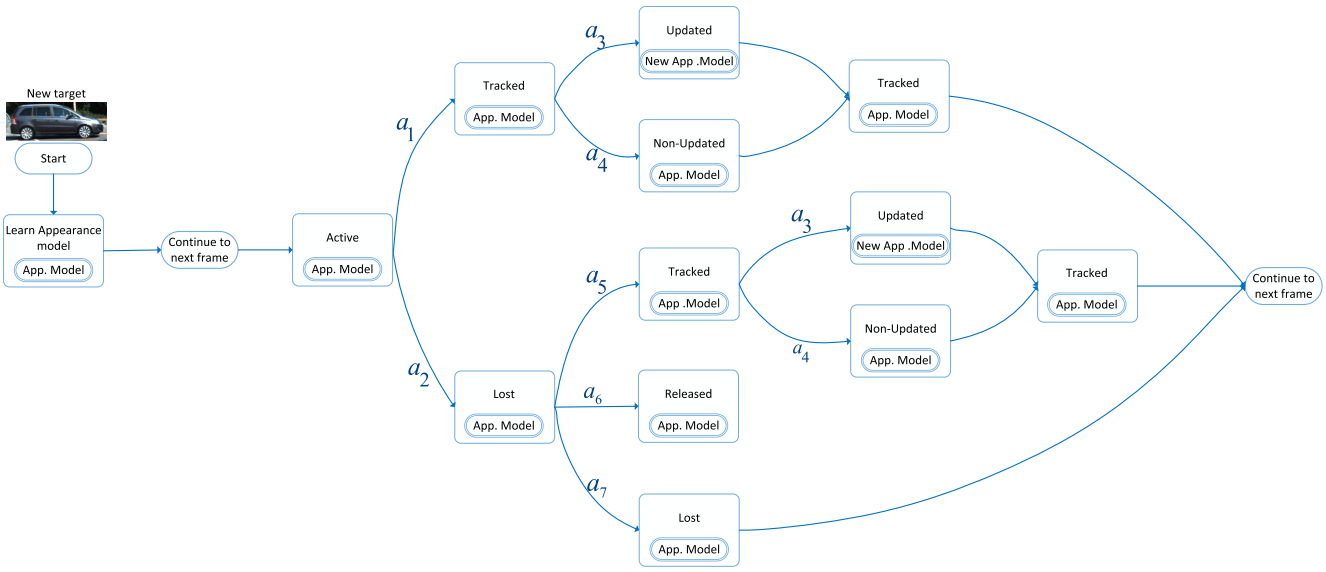


FIGURE 1. The horizontal view of target MDP in our MOT framework.

Deep convolution neural network and the policy gradient based optimization method were used to obtain the optimal policy. Long short term memory (LSTM) [47] was integrated into deep reinforcement learning algorithm to hold temporal information in [45]. Our work is related to [35] and [46]. But Supancic and Ramanan's work [46] was focused on the SOT problem and Xiang *et al.*'s work [35] did not use the DCFB SOT tracker to do MOT. Moreover, no mechanisms have been applied to simultaneously handle occlusion and scale variant in [35] and [46].

III. ONLINE MOT ALGORITHM

As shown in Fig. 1, The overview of our MOT algorithm is presented. The lifetime of each target is modeled as a MDP, and multiple MDPs are exploited to perform MOT.

A. DECISION MAKING

Decision making has drawn great attention due to the development of reinforcement learning. In this section, the combination of DCF and decision making is used for online MOT and the Markov decision process is exploited to model the lifetime of target.

1) MARKOV DECISION PROCESS

In the decision making process, a Markov decision process (MDP) is defined by the tuple $(\mathcal{S}, \mathcal{A}, \Pr(\cdot), R(\cdot))$, where

- $s \in \mathcal{S}$ represents the current state. $\mathcal{S}$ is the current state space.
- The action $a \in \mathcal{A}$ can be taken in the current state $s \in \mathcal{S}$. $\mathcal{A}$ is the current action space.
- The state transition probability $\Pr(\cdot)$ indicates the probability of taking the action $a \in \mathcal{A}$ to achieve the next state in the current state $s \in \mathcal{S}$.
- The reward r can be achieved when the action a is taken in the state s, namely, $r = R(s, a)$.

a: STATES

In reinforcement learning, states represent all informations which are useful for decision making. In our work, the lifetime of a target is regarded as a MDP. Due to the combination of DCF and decision making for online MOT, the appearance model of target and other state informations, such as the location of target, the similarity of target and predicted target, etc, together construct states of MDP. The state space is divided into six subspaces, i.e., $S = S_{active} \cup S_{tracked} \cup S_{lost} \cup S_{updated} \cup S_{non-updated} \cup S_{released}$. Fig. 1 shows the horizontal view of the state transition for those six state subspaces. For every new target, we learn the appearance model of the new target, and then in the next frame, the appearance model is actived. In the active state, the target can transfer to the tracked or lost state. It indicates that the target is tracked or lost. For the lost state, we adjust detections within the region of interest as the same size of the target and then associate the target with these detections, decide which state should be transferred. For the tracked state, it can transfer to the updated or non-updated state that determines whether the appearance model of target is updated or not. Then, it continues to the next frame of video and circulates these state transitions as described above.

b: ACTION, TRANSITION FUNCTION AND REWARD FUNCTION

The action space is showed in Fig. 1. In our work, the policy is a deterministic. It means that in a certain state, the action being took and the next state being transferred are deterministic. For the reward function, the inverse reinforcement learning [48] is exploited to learn it from the ground truth trajectories of targets.

2) POLICY

In reinforcement learning, the policy is a mapping from the state space to the optimal action space. In our work, the policy aims for choosing the optimal actions in those six states. A binary support vector machines (SVM) [49] is trained to learn the reward function, given as

$$R(s, a) = y(a)\left(w^T \phi(s) + b\right) \tag{1}$$

where $\phi(s)$ is the feature vector for decision making that the detail is described in section feature representation; (w, b) are the parameters of our reward function, and $y(a)$ is determined by the action a. In the active, tracked and lost state, they have their own reward function which determines the action $a \in \mathcal{A}$ is taken and what the immediate reward is. The details are stated as follows.

a: POLICY IN THE ACTIVE STATE

As shown in Fig. 1, when the appearance model of target is learned and continue to track, the target chooses an action to play and then transfers to the tracked or lost state. Based on eq. (1), we define that if the action $a = a_1$, $y(a) = 1$, and if the action $a = a_2$, $y(a) = -1$. In this case, we can learn the reward function in the active state through the ground truth trajectories of targets. That is a form of the inverse reinforcement learning [48]. $\phi(s)$ is the extracting feature to train the reward function. The feature vector contains five correlation responses, five normalized cross correlation(NCC) values. The overlap among the detection bounding boxes with tracking results in the bounding box. The details for the feature representation can be seen in the subsection 3.

b: POLICY IN THE TRACKED STATE

The DCFB tracking method has to collect and label training samples by itself. It often collects corrupted training samples, which result in target drift to these corrupted samples. Thus, in the tracked state, the target needs to decide whether the appearance model is updated or not. Intuitively, if the training sample is not corrupted, we should update the appearance model of the target; otherwise, the appearance model does not need to be updated. Similar to the policy in the active state, we also learn the reward function shown in eq. (1). We define that if the action $a = a_3$, $y(a) = 1$, and if the action $a = a_4$, $y(a) = -1$. $\phi(s)$ in the tracked state aims for judging whether the predicted target is reliable or not. Thus, we collect a 10 dimensions feature vector that contains five correlation responses and five NCC values. The details for those features are discussed in the subsection 3.

c: POLICY IN THE LOST STATE

In the active state, occlusion makes target transfer to the lost state. Moreover, in the road scenarios, scale variant and other distractions also make tracking algorithm lose the target and then transfers to the lost state. In order to alleviate these problems, in the lost state, we resize the detection bounding box within the region of interest as the same size as the target and then compute correlation responses and NCC values between them. The reward function is the same as that in the active and tracked state, except for the feature vector and $y(a)$. In the lost state, $y(a) = 1$ if the action $a = a_5$, and $y(a) = -1$ if the action $a = a_7$. Furthermore, the target is decided to transfer to the released state through a certain threshold evaluation.

3) FEATURE REPRESENTATION

As above-mentioned, we try to learn the reward functions in the active, tracked and lost state. In order to well learn these reward function, some effective features are required to extract for training our reward function shown in eq. (1). In the active state, the target decides to choose action a_1 or a_2 and then transfers to the tracked or lost state. A DCFB single object tracker is used to track targets since its correlation response is naturally utilized for MOT.

The correlation responses of DCFB SOT tracker are naturally utilized as a part of features. We also calculate NCC to achieve the similarly of the target with the predicted target. In order to tackle occlusion and scale variant, the correlation responses and NCC values among the upper half, lower half, left half and right half part of the target with the corresponding part of the predicted target are collected. Moreover, the overlap of the detection bounding box and the predicted target bounding box is useful for decision making in the active state. $\phi_1 \sim \phi_{11}$ in Table 1 are features to be extracted for training

TABLE 1. Feature representation for policy learning.

Feature type	Notation	Feature description
Responses	$\phi_1, \phi_2, \phi_3, \phi_4, \phi_5$	The maximum correlation responses among the upper half, lower half, left half, right half and entire part of target with the corresponding part of the predicted target.
NCC	$\phi_6, \phi_7, \phi_8, \phi_9, \phi_{10}$	The NCC values among the upper half, lower half, left half, right half and entire part of target with the corresponding part of the predicted target.
Overlap	ϕ_{11}	The overlap of the detection bounding box and the predicted target bounding box.
Aspect ratio	ϕ_{12}	Ratio in the bounding box height of detection bounding box and the predicted target bounding box.
	ϕ_{13}	Ratio in the bounding box aspect of detection bounding box and the predicted target bounding box.
Score	ϕ_{14}	Normalized detection score.
Distance	ϕ_{15}	Euclidean distance between the center of predicted target bounding box and the bounding box center of target that is predicted by a linear velocity motion model.

the reward function in the active state. We also need to decide whether to use the predicted target to update the appearance model of target or not. In the ideal situation, if the target is not occluded, we should update the appearance model of target. In order to judge whether the target is occluded, $\phi_1 \sim \phi_{10}$ in Table 1 are collected. In order to address scale variant and occlusion, in the lose state, it is necessary to adjust the detection bounding box as the same size as the target and then calculate the correlation responses and NCC values of the target and detections. The aspect ratio provides shape information between the target and the predicted target which is useful in lost state. Moreover, Euclidean distance between the center of the predicted target with the center of the bounding box that is predicted by a linear velocity motion model is also useful for providing the temporal information about the target. It is robust to the appearance change of the target. So in lost state, $\phi_1 \sim \phi_{15}$ except ϕ_{11} in Table 1 are collected.

B. DCFB TRACKING ALGORITHM

One of our novelties is that the DCFB tracking method is integrated into MOT. In order to realize this integration, A DCFB SOT tracker is used for MOT. Moreover, two DCFB trackers with different update frequencies are used to make DCFB tracker more robust to occlusion. The part-based method and a novel update strategy of the appearance model are proposed to improve the robustness of the algorithm to scale variant.

1) DCFB SOT

The DCFB single object tracker learns the appearance model of the target by using the ridge regression method [9]. Specifically, the appearance model $\mathbf{w}$ is learned to minimize the square error over the response map of the correlation filter with samples $\mathbf{x}$ and the expected regression response map $\mathbf{y}$ is given as

$$\min_{\mathbf{w}} \sum_{i} \left(\mathbf{w}^T \mathbf{x}_i - \mathbf{y}_i\right)^2 + \lambda \|\mathbf{w}\|^2 \tag{2}$$

where λ is the regularization parameter for avoiding overfitting and T denotes transpose operator. The problem given by eq. (2) has a simple closed-form solution

$$\mathbf{w} = \left(\mathbf{X}^H \mathbf{X} + \lambda \mathbf{I}\right)^{-1} \mathbf{X}^H \mathbf{y} \tag{3}$$

where H denotes Hermitian transpose operator and $\mathbf{X}^H$ is the Hermitian transpose of $\mathbf{X}$.

Combining with the cyclic shifts, we can simplify the linear regression given by eq. (3), given as

$$\hat{\mathbf{w}} = \frac{\hat{\mathbf{x}}^* \odot \hat{\mathbf{y}}}{\hat{\mathbf{x}}^* \odot \hat{\mathbf{x}} + \lambda} \tag{4}$$

where $\wedge$ denotes discrete fourier transform (DFT) and symbol $*$ denotes complex-conjugate operator. The operation $\odot$ denotes the Hadamard product.

The mode is need to update for continuing tracking the target. The appearance model is updated as

$$\mathbf{A}_t = (1-\mu)\,\mathbf{A}_{t-1} + \mu \hat{\mathbf{x}}^* \odot \hat{\mathbf{y}} \tag{5}$$

$$\mathbf{B}_t = (1-\mu)\,\mathbf{B}_{t-1} + \mu \hat{\mathbf{x}}^* \odot \hat{\mathbf{x}} \tag{6}$$

$$\hat{\mathbf{w}} = \frac{\mathbf{A}_t}{\mathbf{B}_t + \lambda} \tag{7}$$

where μ is the learning rate and t is the frame index. Collecting an image patch $\mathbf{z}$ from the next frame, we can calculate the correlation response map by

$$\mathbf{r} = F^{-1}\left(\hat{\mathbf{w}} \odot \hat{\mathbf{z}}^*\right) \tag{8}$$

where F^{-1} denotes the inverse fast fourier transform (IDFT) transform.

In the DCFB SOT algorithm, the location of the predicted target is determined by eq. (8). Specially, it takes the location of the maximum value in correlation response map as the center of the predicted target. The width and height of the predicted target are the same as the target template. In order to address the conflict problem of update frequency, we use two DCFB trackers with different update frequencies for the tracking. Specifically, one DCFB tracker updates the appearance model conservatively and the other updates the model aggressively. More specifically, in the tracked state, if the target decides to take the action a_3 and then transfers to the updated state, the appearance model of the aggressive DCFB tracker is immediately updated. For the conservative DCFB tracker, the appearance model of it is updated only when the model of the aggressive DCFB tracker are updated more than K times. The aggressive DCFB tracker is sensitive to the dramatic appearance change of the target while the conservative DCFB tracker is robust to the appearance change of target. However, due to the aggressive update of the correlation filter of the aggressive DCFB tracker, the maximum response in correlation response map of the aggressive DCFB tracker is always larger than the maximum response in the correlation response map of the conservative DCFB tracker. Thus, we combine the correlation response map with the overlap value between the tracking result bounding box and detections shown in eq. (9) for determining the final tracking result. Eq. (9) is motivated by the fact that when the aggressive DCFB tracker gradually drifts to the false alarm, although the maximum response of correlation response map is large, but the IOU overlap of the tracking result bounding box and the detection bounding box is small or none. However, for the conservative DCFB tracker, the IOU overlap of the tracking result bounding box and the detection bounding box is large. In this case, if we choose appropriate weight parameters for eq. (9), we can make our tracking result more robust to drifting caused by occlusion. Moreover, if the IOU overlap of the predicted tracking result bounding box and the detection bounding box is larger than a certain threshold, we adapt the detection bounding box as the final predicted location of the target. And we use the final predicted target to update the appearance model of the target. It makes our method more robust to the scale variant.

$$f = \gamma \max(r) + \eta \max(o(t,d)) \tag{9}$$

where γ and η are the weight parameters. r is the correlation response map; t is the tracking result bounding box of the

DCFB tracker; d is the detection bounding box and o denotes the IOU overlap of two bounding boxes.

Due to the frequent scale variant in autonomous driving scenarios, the update strategy for the appearance model of target is also different from the common way as shown in eq. (5)-(7). We follow eq. (4) to re-init the appearance model of the target when the target decides to update the appearance model. In this case, it can well fit the scale change of the target.

All above-mentioned measures make our DCFB tracking algorithm more robust to occlusion and scale variant.

2) PART-BASED METHOD

Some features are extracted by using the part-based method for decision making. As shown in Fig. 2, we partition a target into four sub-block, namely, the upper half, lower half, left half and the right half part.

(a) (b)

FIGURE 2. **(*a*) Partitioning a target into the left half and right half part. The black horizontal line is the separator line. (*b*) Partitioning a target into the upper half and lower half part. The black vertical line is the separator line.**

In order to tackle occlusion and interaction among targets, we learn the appearance models of the upper half, lower half, left half and the right half part of targets by using eq. (4), and then collect the maximum responses of the correlation response maps. Thus, we can obtain the corresponding four correlation responses and use them as a part of features to do decision making. Considering the frequent scale change in MOT scenarios especially in the road scenarios, we resize the upper half, lower half, left half, right half and entire part of the target to the fixed width and height $[w, h]$. Then, we compute NCC values between the corresponding part of the target with the predicted target. Resizing and NCC values make our algorithm more robust to scale variant in the road scenarios.

C. ONLINE MOT

In this section, the learning of the reward function in the inverse reinforcement learning fashion and the pipeline about how to organize multiple MDPs to do MOT are summarized.

1) INVERSE REINFORCEMENT LEARNING

In this paper, we train a binary SVM in the inverse reinforcement learning to learn the reward function. We hypothesize that there are $V = \{v_n\}_{n=1}^{N}$ video sequences and there are $T_m = \{t_{m'n}\}_{m'=1}^{M'_n}$ ground truth targets in video v_n. As shown in Fig. 1, we aim to learn reward functions in active, tracked and lost state and then accurately track all targets and seasonably update the appearance model. At the first, we init the reward functions in the active, tracked and lost state with the weights (w_0, b_0). For the active state, we have the initial training set $S_{active} = \emptyset$. For the tracked state and lost state, we have the initial training set $S_{tracked} = \emptyset$ and $S_{lost} = \emptyset$ respectively. As described above, we init policies in the active, tracked and lost state with the weights (w_0, b_0). Thus, we can make a decision with the corresponding policy in the active, tracked and lost state. If we make an appropriate decision which is the same as the ground truth targets, there is nothing to be done. Otherwise, we collect features into the corresponding training set and then update the corresponding policy. For example, in the active state, there are two situations that make us update the corresponding policy, e.g., (1) for the target $t_{m'n}$, if the policy decides to take the action a_1 but the target $t_{m'n}$ is covered; (2) if the policy decides to take action a_2 but the target $t_{m'n}$ is visible. The above two situations make an inappropriate decision so that we should add features $\phi_1 - \phi_{11}$ and the corresponding label $y(a_1) = 1$ or $y(a_2) = -1$ to the training set. There exist similar operations in the tracked and lost state. The details can be seen in Algorithm 1 given in the Appendix A.

2) MOT WITH MDPs

As shown in Algorithm 1, we learn the policies or the reward functions of MDPs. Thus, in this section, we focus to how to apply MDPs and DCFB method for the MOT problems. When a new target appears, we model the lifetime of the target as a MDP. Then, the target is in the active state and transfers to the tracked or lost state. If the target transfers to the tracked state, the target decides to transfer to the updated or non-updated state. For the lost state, detections within the region of interest should be resized as the same size of the target and then features are extracted to feed into the reward function for deciding whether the target transfers to the tracked state or the lost state. When the target loses for certain times, the MDP of the target is released. Moreover, a number of targets appear on the same frame of a video, we should take actions to organize them and address the conflict problem that different targets are predicted to be in the same positions. We compare the maximum correlation responses of correlation response maps of the different targets, and the tracking priorities of different targets are determined by the magnitude of the maximum correlation responses. Specially, the target with a large maximum correlation response have a higher priority. The target in the tracked state has a higher priority than the target in lost state. Moreover, if the tracking result bounding boxes of different targets have a large IOU overlap which is larger than a certain threshold, we perform the non-maximum suppression (NMS) according to the maximum correlation responses of the targets. The nature property of the DCFB tracker, correlation response, can be used in MOT naturally. The details of our MOT algorithm is described in Algorithm 2 given in the Appendix B.

IV. EXPERIMENTS

A. DATASETS

We evaluate our MOT framework based on the famous KITTI [1] tracking benchmark for autonomous

TABLE 2. Tracking results on the test dataset from the KITTI tracking benchmark.

	MOTA	MOTP	Recall	Precision	MT	PT	ML	TP	FP	IDS	FRAG
MDP[35]	76.59	82.10	80.26	98.00	56.31	34.46	8.46	29747	606	130	387
LP-SSVM[52]	77.63	77.80	83.35	96.27	56.31	35.23	8.46	31997	1239	62	539
NOMT[30]	78.15	79.46	83.23	96.78	57.23	29.54	13.38	31854	1061	31	207
MCMOT-CPD[53]	78.90	82.12	81.84	98.97	52.31	36.00	11.69	30247	316	228	536
RRC-IIITH[33]	84.24	85.73	88.80	97.95	73.23	24.00	2.77	33656	705	468	944
OURS	86.75	85.36	89.63	98.64	73.23	23.08	3.69	33923	466	169	469

TABLE 3. Tracking results on the KITTI tracking test dataset using RRC detection results.

	MOTA	MOTP	Recall	Precision	MT	PT	ML	TP	FP	IDS	FRAG
RRC-IOU[32]	73.85	79.05	78.28	97.72	38.62	46.92	14.46	29388	687	151	672
RRC-MDP[35]	81.13	84.43	84.71	98.12	57.69	29.23	13.08	31119	596	275	544
RRC-IIITH[33]	84.24	85.73	88.80	97.95	73.23	24.00	2.77	33656	705	468	944
RRC-OURS	86.75	85.36	89.63	98.64	73.23	23.08	3.69	33923	466	169	469

TABLE 4. Ablation study on features for policy learning.

	MOTA	MOTP	Recall	Precision	MT	PT	ML	TP	FP	IDS	FRAG
w/o NCC and response	54.24	84.71	58.18	98.86	22.62	47.38	30.00	321421	247	97	562
w/o NCC	73.85	79.05	78.28	97.72	38.62	46.92	14.46	29388	687	151	672
w/o response	81.13	84.43	84.71	98.12	57.69	29.23	13.08	31119	596	275	544
w/o distance	83.63	85.57	86.99	98.51	66.46	25.85	7.69	32872	497	215	527
w/o aspect ratio	83.66	85.73	85.97	99.23	62.15	30.15	7.69	32383	250	81	449
with all features	86.75	85.36	89.63	98.64	73.23	23.08	3.69	33923	466	169	469

TABLE 5. Tracking results on the KITTI tracking dataset with different update frequency of the appearance model.

	MOTA	MOTP	Recall	Precision	MT	PT	ML	TP	FP	IDS	FRAG
K=0	84.86	85.52	87.79	98.97	67.38	27.69	4.92	33124	344	254	712
K=5	86.24	85.47	88.76	98.99	71.38	23.69	4.92	33504	343	149	444
K=10	86.75	85.36	89.63	98.64	73.23	23.08	3.69	33923	466	169	469
K=15	86.27	85.28	90.08	98.03	74.77	21.23	4.00	34097	686	281	546

TABLE 6. Tracking results on the KITTI tracking dataset with different weight parameters in equation 9.

	MOTA	MOTP	Recall	Precision	MT	PT	ML	TP	FP	IDS	FRAG
γ= 1,$\eta = 0.5$	80.31	85.29	88.76	95.18	71.08	24.31	4.62	33589	1700	819	1125
γ= 1,$\eta = 1$	86.40	85.19	89.64	98.40	73.85	22.31	3.85	33917	552	207	522
γ= 0.5,$\eta = 1$	86.75	85.36	89.63	98.64	73.23	23.08	3.69	33923	466	169	469

driving scenarios. The KITTI tracking dataset contains 21 training and 29 testing video sequences of road scenarios. The training sequences are annotated and the ground truth is released, Thus, we use the ground truth trajectories of targets to learn the reward function in inverse reinforcement learning fashion. Since our MOT algorithm is in the tracking-by-detection fashion, we need to use a detection algorithm for detecting objects in each frame of video sequences. In this case, in order to compare with other state-of-the-art MOT algorithms, we use the recurrent rolling convolution (RRC) [51] algorithm to detect objects in KITTI tracking dataset and we report our tracking results on the car class.

B. EVALUATION METRICS

In order to evaluate the performance of our MOT algorithm, we exploit the popular CLEAR MOT metrics [50]. It contains multiple object tracking precision (MOTP), multiple object tracking accuracy (MOTA), the number of false negatives(FN), the number of false positives (FP), the percentage of the mostly track targets (MT, percentage of the overlap of ground truth objects and tracking result bounding box larger than 80%), the percentage of the mostly lost targets (ML, percentage of the overlap of ground truth objects and tracking result bounding box less than 20%), the number of id switches(IDS) and the number of times that a trajectory is fragmented(Frag).

C. IMPLEMENTATION DETAILS

We present the main steps for the learning of reward functions in Algorithm 1 and the main steps for online MOT in Algorithm 2. We use the HCF algorithm [16] as the base DCFB tracker. The update frequency K of the appearance model in the conservative DCFB tracker is chose as

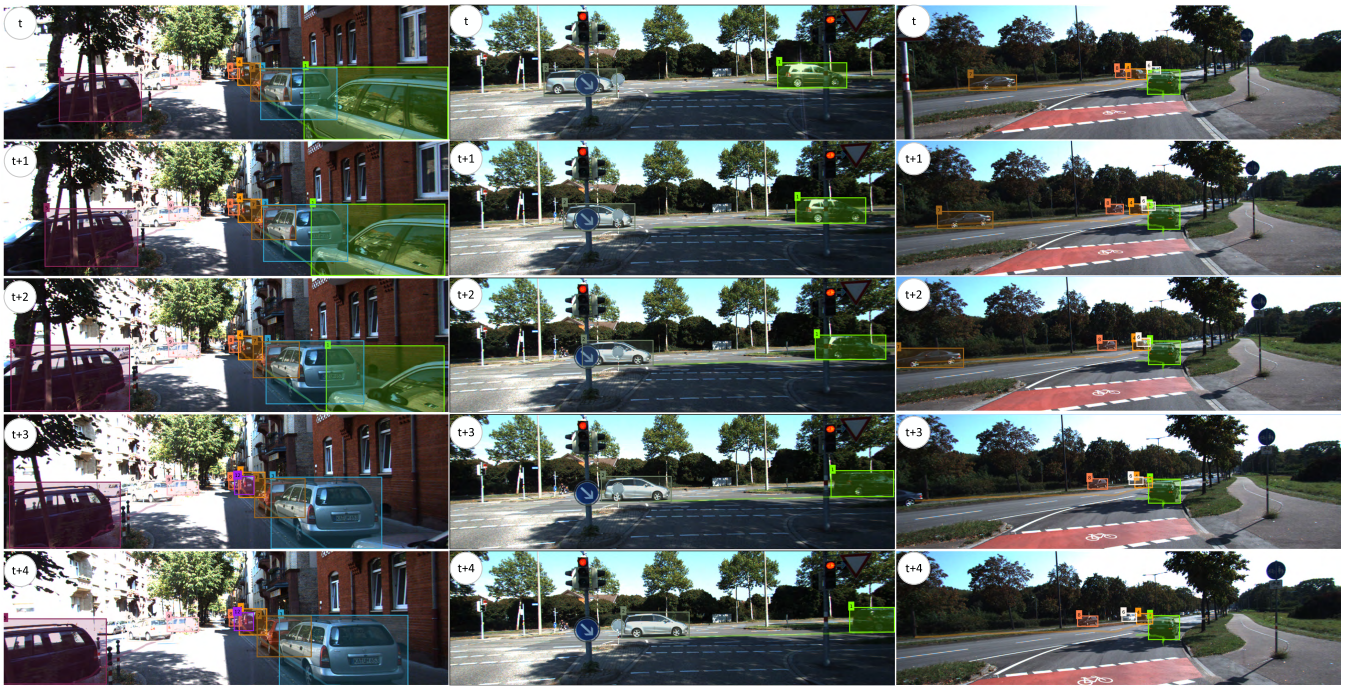


FIGURE 3. Qualitative results of some typical challenging scenarios.

FIGURE 4. Qualitative results of some typical challenging scenarios.

10 and we show the effectiveness of it in Table 5. The result in Table 6 shows that the weight parameters γ and η in eq. (9) are set to be 0.5 and 1 are the optimal selection. We employ the setting in [35] and set $[w, h] = [24, 12]$. In order to compare with other state-of-the-art MOT algorithms fairly, we use RRC [51] algorithm to detect objects, and then use these detections to do MOT and evaluate our algorithm on KITTI tracking benchmark.

D. PERFORMANCE EVALUATION

As shown in Table 2, we compare our approach with other state-of-the-art MOT approaches on the KITTI

Algorithm 1 Learning Policy via Inverse Reinforcement Learning

1) **Input settings:**
video sequences $V = \{v_n\}_{n=1}^{N}$ (N *videos totally*);
object detections $D_m = \{d_{mn}\}_{m=1}^{M_n}$ (M_n *detections in the nth video*);
ground truth targets $T_m = \{t_{m'n}\}_{m'=1}^{M'_n}$ (M'_n *targets in the nth video*);
2) **Initialization:**
policy in active state $w_{active} \leftarrow w_0, b_{active} \leftarrow b_0$;
policy in tracked $w_{tracked} \leftarrow w_0, b_{tracked} \leftarrow b_0$;
policy in lost $w_{lost} \leftarrow w_0, b_{lost} \leftarrow b_0$;
update frequency $K = 0$;
3) **Optimization:**
repeat
for n**=1:N**
foreach traget $t_{m'n}$ **in video** v_n
$f \leftarrow$ the first index of the frame that target $t_{m'n}$ is detected;
learn the appearance model of the upper half, lower half, left half, right half and entire part of the detected object;
initialize MDP to the active state;
while $f \leq$ **the last index of the frame the target appeared**
following the current policy in the active state and taking an action a_{active} in the active state;
achieve the ground truth action a_{active_gt} from the true trajectory;
if $a_{active\_gt} = a_{active}$ **then:**
transfer to the current state s by taken the action a_{active};
$f = f + 1$;
if current state s **is lost state:**
resize detection d_i to the same size as tracking template, and then compute features $\phi_1 \sim \phi_{15}$ except overlap;
following the current policy in the lost state and take a action a_{lost};
achieve the ground truth action a_{lost_gt} from the true trajectory;
if $a_{lost\_gt} = a_{lost}$ **then:**
transfer to the current state s by takeing the action a_{lost};
else:
compute label $y\,(a_{lost})$ and collect features $\phi_1 \sim \phi_{15}$ except ϕ_{11} as shown in table 1;
$S_{lost} \leftarrow S_{lost} \cup \{(\phi_1 \sim \phi_{15}) - \phi_{11}, y\,(a_{lost})\}$;
update policy according to eq. (1);
end if
end if
if current state s **is tracked state:**
following the current policy in the tracked state and take a action $a_{tracked}$ in the tracked state;
achieve the ground truth action $a_{tracked\_gt}$ from the true trajectory;
if $a_{tracked\_gt} = a_{tracked}$ **then:**
transfer to the current state s by takeing the action $a_{tracked}$;
if current state s **is updated state:**
update the appearance model of the aggressive DCFB tracker;
$K = K + 1$;
if $K >= 10$ **then:**
update the appearance model of the conservative DCFB tracker;
$K = 0$;
end if
else:
the current state s is non-updated state;
end if
else:
compute label $y\,(a_{tracked})$ and collect features $\phi_1 \sim \phi_{10}$ as shown in table 1;
$S_{tracked} \leftarrow S_{tracked} \cup \{(\phi_1 \sim \phi_{10})\,, y\,(a_{tracked})\}$;;
update policy according to eq. (1);
end if
else:
compute label $y\,(a_{active})$ and collect features $\phi_1 \sim \phi_{11}$ as shown in Table 1;
$S_{tracked} \leftarrow S_{tracked} \cup \{(\phi_1 \sim \phi_{11})\,, y\,(a_{tracked})\}$;
update policy according to eq. (1);
if current state s **is tracked state:**
break;
end if
end if
if $f >$ **the last index of the frame the target appeared**
mark target $t_{m'n}$ is tracked successfully;
end if
end while
end for
end for
until all targets are tracked successfully or over given iterations

Algorithm 2 Online MOT With MDPs

1) **Input:**
video sequences $V = \{v_n\}_{n=1}^{N}$ (N *videos totally*);
object detections $D_m = \{d_{mn}\}_{m=1}^{M_n}$ (M_n *detections in the nth video*);
policy in the active, tracked and lost state;
2) **output:**
Trajectories of targets $\Gamma = \{t_k\}_{k=1}^{K}$;
2) **Initialization:**
$\Gamma \leftarrow \emptyset$;
$K = 0$;
3) **Optimization:**
for n**=1:N**
foreach frame f **in video** v_n
foreach tracked target t_m **in** Γ
move the MDP of target t_m to the active state;
follow the policy in the active state and transfer to the state s;
if state s **is lost state:**
resize detections within ROI to the same size as tracking template,
and then compute features $\phi_1 \sim \phi_{15}$;
follow the policy in the lost state and transfer to the state s;
end if
if state s **is tracked state:**
follow the policy in the tracked state and transfer to the state s;
if state s **is updated state:**
update the appearance model of the aggressive DCFB tracker;
$K = K + 1$;
if $K >= 10$ **then:**
update the appearance model of the conservative DCFB tracker;
$K = 0$;
end if
else:
the current state s is non-updated state;
end if
end if
end for
foreach lost target t_m **in** Γ
resize detections within ROI to the same size as target template,
and then compute features $\phi_1 \sim \phi_{15}$ except ϕ_{11};
follow the policy in the lost state and transfer to the state s;
if state s **is tracked state:**
follow the policy in the tracked state and transfer to the state s;
if state s **is updated state:**
update the appearance model of the aggressive DCFB tracker;
$K = K + 1$;
if $K >= 10$ **then:**
update the appearance model of the conservative DCFB tracker;
$K = 0$;
end if
else:
the current state s is non-updated state;
end if
end if
end for
foreach detection d_{mn} **not covered by any target**
initialize a MDP for the detection and store it to Γ;
learn the the appearance models of the detection as eq. (3);
sort Γ via correlation response;
end for
end for
end for

tracking benchmark. Before our work, the published MOT approach which has the best performance on KITTI car benchmark is RRC-IIITH [33]. It is seen that our approach outperforms RRC-IIITH by 2.51% in MOTA. Moreover, ID switch of our approach is also less than that of RRC-IIITH. In the tracking-by-detection fashion MOT approach, the detection results are very important for the performance of MOT approaches. Thus, in order to compare our approach with other state-of-the-art MOT algorithms fairly, we use the same detection results for some state-of-the-art MOT approaches which have release their codes. The comparison result is listed in Table 3. Our approach ourperforms these state-of-the-art MOT algorithms with the same detection results.

E. ABLATION STUDY

We design 15 dimension features for our MOT framework as shown in Table 1. A thorough ablation analysis is performed to verify the importance of these features. As shown in Table 4, removing NCC values or responses significantly deteriorates the performance of our algorithm since they can address scale variant and occlusion. Distance and aspect ratio also show their importance to our approach in Table 4. We also show the influence of the update frequency of the appearance model. $K = 10$ is the best choice as shown in table 5, more quick update frequency may make target drift to the corrupted training samples and a small update frequency cannot handle the appearance change of object well. The effect of two DCFB trackers is showed in Table 6.

F. QUALITATIVE RESULTS

In this section, we present qualitative results of some typical challenging sequences in the road scenarios in Fig. 3 and Fig. 4. These qualitative results clearly indicate that our approach can address difficulties especially occlusion and scale variant. For example, the first column of Fig. 3 shows that our approach is robust to scale variant and background clutter. The second, third column of Fig. 3 and the third column of Fig. 4 indicate that our algorithm can well tackle occlusion caused by the traffic sign, interaction among targets and tree. The first and second column of Fig. 4 show the robustness of the out-of-plane rotation and scale variant.

V. CONCLUSION

In this paper, an online MOT approach via combining discriminative correlation filters with making decision was proposed. In order to alleviate the adverse effect of the corrupted training samples in DCFB method, whether to update the appearance model of the target or not was regarded as a state transition in MDP. In order to improve the robust of our algorithm to the scale variant and occlusion, the state transition was also modeled in the lost state. Moreover, two DCFB trackers with different update frequencies and a novel update strategy different from the common way were applied to improve the robust of our algorithm to the occlusion and scale variant. Furthermore, the part-based method was used to extract effective features that help our algorithm to tackle occlusion, interactions among targets and scale variant. The results demonstrated the superiority of our method compared with the existing methods for addressing occlusion and scale variant. Experiment results in challenging tracking dataset, KITTI tracking dataset, verified the efficiency of our proposed MOT algorithm.

APPENDIX A

See Algorithm 1.

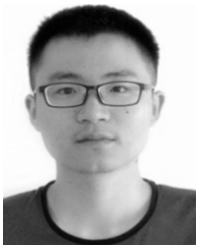
APPENDIX B

See Algorithm 2.

REFERENCES

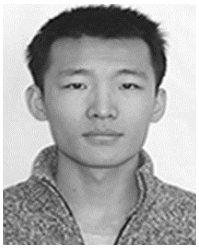
[1] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE CVPR*, Providence, RI, USA, Jun. 2012, pp. 3354–3361.
[2] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE CVPR*, Columbus, OH, USA, Jun. 2013, pp. 2411–2418.
[3] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1442–1468, Jul. 2014.
[4] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proc. IEEE CVPR*, San Francisco, CA, USA, Jun. 2010, pp. 2544–2550.
[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NIPS*, Lake Tahoe, NV, USA, Dec. 2012, pp. 1097–1105.
[6] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556
[7] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE CVPR*, Boston, MA, USA, Jun. 2015, pp. 3431–3440.
[8] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proc. ECCV*, Firenze, Italy, Oct. 2012, pp. 702–715.
[9] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.
[10] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. BWVA*, Nottingham, U.K., Sep. 2014, pp. 1–11.
[11] M. Danelljan, F. S. Khan, M. Felsberg, and J. van de Weijer, "Adaptive color attributes for real-time visual tracking," in *Proc. IEEE CVPR*, Columbus, OH, USA, Jun. 2014, pp. 1090–1097.
[12] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proc. ICCV*, Santiago, Chile, Dec. 2015, pp. 4310–4318.
[13] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking," in *Proc. IEEE CVPR*, Las Vegas, NV, USA, Jun. 2016, pp. 1430–1438.
[14] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, "Staple: Complementary learners for real-time tracking," in *Proc. IEEE CVPR*, Las Vegas, NV, USA, Jun. 2016, pp. 1401–1409.
[15] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *Proc. IEEE ICCV Workshops*, Santiago, Chile, Dec. 2015, pp. 58–66.
[16] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proc. IEEE ICCV*, Santiago, Chile, Dec. 2015, pp. 3074–3082.
[17] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. S. Torr, "End-to-end representation learning for correlation filter based tracking," in *Proc. IEEE CVPR*, Honolulu, HI, USA, Jul. 2017, pp. 5000–5008.
[18] E. Gundogdu and A. A. Alatan, "Good features to correlate for visual tracking," *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2526–2540, May 2018.
[19] F. Li, F. Li, C. Tian, W. Zuo, L. Zhang, and M.-H. Yang. (2018). "Learning spatial-temporal regularized correlation filters for visual tracking." [Online]. Available: https://arxiv.org/abs/1803.08679
[20] Q. Chu, W. Ouyang, H. Li, X. Wang, B. Liu, and N. Yu. (2017). "Online multi-object tracking using CNN-based single object tracker with spatial-temporal attention mechanism." [Online]. Available: https://arxiv.org/abs/1708.02843
[21] M. Danelljan, A. Robinson, F. Khan, and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *Proc. ECCV*, Amsterdam, The Netherlands, Oct. 2016, pp. 472–488.
[22] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ECO: Efficient convolution operators for tracking," in *Proc. IEEE CVPR*, Honolulu, HI, USA, Jul. 2017, pp. 21–26.
[23] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *Proc. ECCV*, Zurich, Switzerland, Sep. 2014, pp. 254–265.
[24] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. NIPS*, Montreal, QC, Canada, Dec. 2015, pp. 91–99.

[25] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE CVPR*, Honolulu, HI, USA, Jul. 2017, pp. 2117–2125.
[26] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Proc. NIPS*, Barcelona, Spain, Dec. 2016, pp. 379–387.
[27] A. Andriyenko, K. Schindler, and S. Roth, "Discrete-continuous optimization for multi-target tracking," in *Proc. IEEE CVPR*, Providence, RI, USA, Jun. 2012, pp. 1926–1933.
[28] J. Berclaz, F. Fleuret, E. Türetken, and P. Fua, "Multiple object tracking using k-shortest paths optimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1806–1819, Sep. 2011.
[29] A. Milan, S. Roth, and K. Schindler, "Continuous energy minimization for multitarget tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, pp. 58–72, Jan. 2014.
[30] W. Choi, "Near-online multi-target tracking with aggregated local flow descriptor," in *Proc. IEEE ICCV*, Santiago, Chile, Dec. 2015, pp. 3029–3037.
[31] S. Schulter, P. Vernaza, W. Choi, and M. Chandraker, "Deep network flow for multi-object tracking," in *Proc. IEEE CVPR*, Honolulu, HI, USA, Jul. 2017, pp. 6951–6960.
[32] E. Bochinski, V. Eiselein, and T. Sikora, "High-Speed tracking-by-detection without using image information," in *Proc. AVSS*, Lecce, Italy, Aug. 2017, pp. 1–6.
[33] S. Sharma, J. A. Ansari, J. K. Murthy, and K. M. Krishna. (2018). "Beyond pixels: Leveraging geometry and shape cues for online multi-object tracking." [Online]. Available: https://arxiv.org/abs/1802.09298
[34] L. Leal-Taixé, C. Canton-Ferrer, and K. Schindler, "Learning by tracking: Siamese CNN for robust target association," in *Proc. IEEE CVPR Workshops*, Las Vegas, NV, USA, Jun. 2016, pp. 33–40.
[35] Y. Xiang, A. Alahi, and S. Savarese, "Learning to track: Online multi-object tracking by decision making," in *Proc. IEEE ICCV*, Santiago, Chile, Dec. 2015, pp. 4705–4713.
[36] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proc. IEEE CVPR*, New York, NY, USA, Jun. 2006, pp. 1815–1821.
[37] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah, "Part-based multiple-person tracking with partial occlusion handling," in *Proc. IEEE CVPR*, Providence, RI, USA, Jun. 2012, pp. 1815–1821.
[38] L. Xu, W. Li, H. Wu, and Q. Li, "Online multi-object tracking based on global and local features," in *Proc. IEEE VCIP*, Chengdu, China, Nov. 2016, pp. 1–6.
[39] T. Liu, G. Wang, and Q. Yang, "Real-time part-based visual tracking via adaptive correlation filters," in *Proc. IEEE CVPR*, Boston, MA, USA, Jun. 2015, pp. 4902–4912.
[40] D. Silver *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.
[41] D. Silver *et al.*, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.
[42] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, 2015.
[43] S. Khim, S. Hong, Y. Kim, and P. K. Rhee, "Adaptive visual tracking using the prioritized Q-learning algorithm: MDP-based parameter learning approach," *Image Vis. Comput.*, vol. 32, no. 12, pp. 1090–1101, Dec. 2014.
[44] S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Y. Choi, "Action-decision networks for visual tracking with deep reinforcement learning," in *Proc. IEEE CVPR*, Honolulu, HI, USA, Jul. 2017, pp. 1349–1358.
[45] D. Zhang, H. Maei, X. Wang, Y.-F. Wang. (2017). "Deep reinforcement learning for visual object tracking in videos." [Online]. Available: https://arxiv.org/abs/1701.08936
[46] J. Supančič, III, and D. Ramanan, "Tracking as online decision-making: Learning a policy from streaming videos with reinforcement learning," in *Proc. IEEE ICCV*, Venice, Italy, Oct. 2017, pp. 322–331.
[47] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Aug. 1997.
[48] A. Y. Ng and S. J. Russell, "Algorithms for inverse reinforcement learning," in *Proc. ICML*, Stanford, CA, USA, Jun. 2000, pp. 663–670.
[49] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. 5th Annu. Workshop Comput. Learn. Theory*, Pittsburgh, PA, USA, Jul. 1992, pp. 144–152.
[50] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *EURASIP J. Image Video Process.*, vol. 2008, pp. 246–309, Dec. 2008.
[51] J. Ren *et al.*, "Accurate single stage detector using recurrent rolling convolution," in *Proc. IEEE CVPR*, Honolulu, HI, USA, Jul. 2017, pp. 752–760.
[52] S. Wang and C. C. Fowlkes, "Learning optimal parameters for multi-target tracking with contextual interactions," *Int. J. Comput. Vis.*, vol. 12, no. 3, pp. 484–501, Mar. 2017.
[53] 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 *Proc. IEEE ECCV Workshops*, Amsterdam, The Netherlands, Oct. 2016, pp. 68–83.



CHENGLONG WU received the B.Sc. degree from Xidian University, Xi'an, China, in 2015. He is currently pursuing the Ph.D. degree with the Institute of Electronics, Chinese Academy of Sciences, Beijing, China.

His research interests include deep learning, reinforcement learning, and computer vision, especially on object detection and object tracking.



HAO SUN received the B.Sc. degree from the Beijing Institute of Technology, Beijing, China, in 2007, and the M.Sc. and Ph.D. degrees from the Institute of Electronics, Chinese Academy of Sciences, Beijing, in 2009 and 2012, respectively.

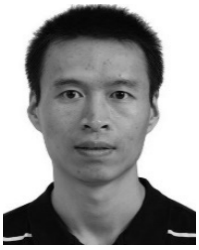
He is currently an Associate Professor with the Institute of Electronics, Chinese Academy of Sciences. His research interests include computer vision, pattern recognition, and remote-sensing image processing.

HONGQI WANG received the B.Sc. degree from the Changchun University of Science and Technology, Changchun, China, in 1983, the M.Sc. degree from the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun, in 1988, and the Ph.D. degree from the Institute of Electronics, Chinese Academy of Sciences, Beijing, China, in 1994.

He is currently a Professor with the Institute of Electronics, Chinese Academy of Sciences. His research interests include computer vision and remote-sensing image understanding.

KUN FU received the B.Sc., M.Sc., and Ph.D. degrees from the National University of Defense Technology, Changsha, China, in 1995, 1999, and 2002, respectively.

He is currently a Professor with the Institute of Electronics, Chinese Academy of Sciences, Beijing, China. His research interests include computer vision, remote-sensing image understanding, geospatial data mining, and visualization.



GUANGLUAN XU received the B.Sc. degree from Beijing Information Science and Technology University, Beijing, China, in 2000, and the Ph.D. degree from the Institute of Electronics, Chinese Academy of Sciences, Beijing, in 2005.

He is currently a Professor with the Institute of Electronics, Chinese Academy of Sciences. His research interests include remote-sensing image understanding and geospatial data mining and visualization.

WENKAI ZHANG received the B.Sc. degree from the China University of Petroleum, Qingdao, China, in 2013. He is currently pursuing the Ph.D. degree with the Institute of Electronics, Chinese Academy of Sciences, Beijing, China.

His research interests include deep learning and computer vision, especially on object detection and image caption.

XIAN SUN received the B.Sc. degree from Beihang University, Beijing, China, in 2004, and the M.Sc. and Ph.D. degrees from the Institute of Electronics, Chinese Academy of Sciences, Beijing, in 2006 and 2009, respectively.

He is currently an Associate Professor with the Institute of Electronics, Chinese Academy of Sciences. His research interests include computer vision and remote-sensing image understanding.