Cross-View Contextual Relation Transferred Network for Unsupervised Vehicle Tracking in Drone Videos

Wenfeng Song[1], Shuai Li[12*], Tao Chang[1], Aimin Hao[1], Qinping Zhao[1], and Hong Qin[3]*
1 State Key Laboratory of Virtual Reality Technology and Systems, Beihang University,
2 Beihang University Qingdao Research Institute, 3 Stony Brook University, USA.
{songwenfeng,lishuai*,changtao,ham,zhqp@buaa.edu.cn}, qin@cs.stonybrook.edu
*

Abstract

Recently CNN-centric object tracking methods have been gaining tremendous success in ground-view videos, however, it remains hard to cope with vehicle tracking in unmanned aerial vehicle (UAV) videos. The key difficulties mainly stem from lacking large-scale well-labeled training datasets and view-invariant appearance model for fast-moving drone-view vehicles. We enhance the vehicle's cross-view feature by exploring relations between the pivotal context and the target to facilitate unsupervised vehicle tracking. The relation is modeled as the relevance of the target and its contextual regions in the tracking task. Specifically, we propose a contextual relation actor-critic (CRAC) framework integrates an actor-critic agent with a dual GAN learning mechanism, which aims to dynamically search the related contextual regions and transfer the relations from ground-view to drone-view videos while retaining the discriminative features. We demonstrate that CRAC could be applied to several state-of-the-art trackers by extensive experiments and ablation studies on four public benchmarks. All the experiments confirm that, our CRAC can improve the performance of state-of-the-art methods in terms of accuracy, robustness, and versatility.

1. Introduction and Motivation

With the built-in advantages of high mobility and flexibility, UAVs/drones are becoming more prominent in security surveillance and rescue-relevant applications in recent years. Drone-view videos can capture large-scale scene information more efficiently and conveniently than the ground-view ones. Despite the widespread tracking algorithms/techniques in computer vision field, it remains a challenging task for the accurate tracking in drone-view videos.

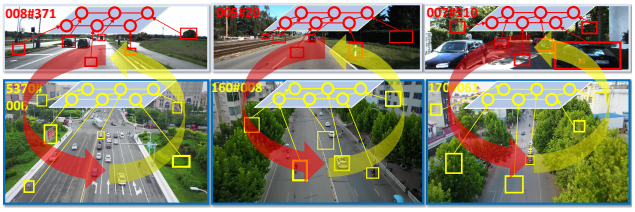

Figure 1. Contextual relations are stable across views. For example, vehicles are more likely to appear on the road instead on the tree. Red indicates the relations in KITTI dataset and yellow indicates the relations in Drone2018 dataset.

Two significant challenges hinder the drone-view tracking. First, the drone-view videos unavoidably leads to largely changed vehicle appearances compared to the ground-view videos due to different camera angles, relative velocity, and the intermittent top-view occlusions. Second, the common obstacle for the deep learning methods is the heavy dependence on a large amount of labeled data. However, it is impossible to collect training datasets for all possible types of scenarios and vehicles. Besides, deep learning methods pre-trained on existing vehicle datasets are hard to be directly adapted to new scenario.

To alleviate, unsupervised tracking methods are proposed, which aims to continuously identify and locate the targeted vehicles without well annotated videos. Existing unsupervised transferring methods provide a viable approach for nature image recognition and segmentation [33, 23, 10], but the large appearance variation from ground view to drone view makes it extremely hard to define universal features across different views, thus new improvements are in urgent demand for the unsupervised cross-view transfer learning.

In terms of the largely changed appearance, the relation built on the target and surrounding regions provides clues

*Shuai Li and Hong Qin are corresponding authors. This research is supported in part by National Key R&D Program of China (NO. 2018YFB1700603), National Natural Science Foundation of China (NO. 61672077 and 61532002), Beijing Natural Science FoundationHaidian Primitive Innovation Joint Fund (L182016), National Science Foundation of USA (NO. IIS-0949467, IIS-1047715, IIS-1715985, and IIS-1049448).

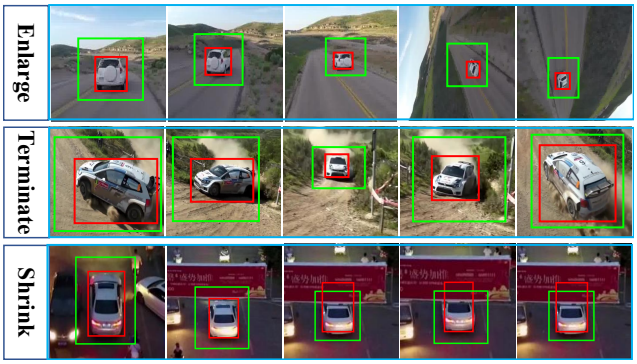

Figure 2. The contextual relations of CRAC. Green boxes: context window. Red boxes: predicted target boxes. (1) Enlarge: the actions tend to find more clues in neighboring regions (e.g., trees, traffic lanes); (2) Shrink: the actions tend to remove the confusing and noisy parts in neighboring regions (e.g., occlusions, other vehicles); (3) Terminate: the current context is good enough to achieve good performance.

for the partially occluded parts and largely changed appearance (e.g., road and trees deformation), as shown in Fig. 1. Furthermore, the tracking task is to learn a discriminative boundary between target and background. Therefore, the relation between target and its neighboring context is the essential element for tracking network. Existing trackers rarely model the contextual relations due to the uncertainty and complexity of the moving target. Instead, we model the contextual relations as the relevance between contextual region and tracking target. The relation is affected by the size of the contextual regions around the target: too small size of contextual region around the currently predicted location can not provide sufficient appearance clues, too large size of contextual region may bring the noisy clues irrelevant with the tracking task (in Fig. 2). Therefore, we dynamically derive the size with an actor-critic agent under the guidance of the tracking performance.

In terms of tracking the never-before-seen drone-view vehicles without dense annotations, we find the contextual relations are more universal than single objects' appearance features across views (e.g., low resolution, aspect ration changes). As shown in Fig. 1, even though the object scales and views are totally different in the KITTI and Drone2018 datasets, the involved vehicle-background relations appear to be stable in different views. Hence, to better adapt to the drone views, we propose a dual GAN learning mechanism consisting of a tracking-guided CycleGAN [38] (T-GAN) and an attention GAN (A-GAN). This mechanism bridges the gaps of appearance from drone view to ground view. Enabled by this new mechanism, the contextual relations are transferred by T-GAN, and further refined from local contextual region to global image range by A-GAN. The salient contributions of this paper can be summarized as follows.

- We propose to efficiently model the contextual relation as the relevance between the target and its contextual regions. We further design an actor-critic agent to dynamically make decisions on contextual relation for the certain vehicle target to the largely changed appearance feature across views.

- We propose a dual generative adversarial (GAN) learning mechanism to transfer the contextual relations across views, which is dedicated to the effective transfer of cross-view appearance features and refine it with the generated attention map from local to global.

- We propose a unified contextual relation actor-critic (CRAC) framework to seamlessly integrate the dynamic context search with the dual GAN transfer. The framework makes the actor-critic agent interactively update the feedback from the dual GANs embedded tracking network in an unsupervised way.

2. Brief Background Review

Visual Tracking. Visual tracking has undergone extensive studies over several decades.

Some state-of-the-art trackers [13] train the tracking task as a binary classification on a discriminative boundary between the tracking target and its background, which is referred as tracking-by-detection, such as MDNET [21], FCNT [34], VITAL [29]. Some recent works [37, 30, 24, 3, 12] attempt to utilize the reinforcement learning to locate the tracking target. However, this kind of methods need numerous proposals, which leads to high computation cost.

Recently, to speed up the trackers, , siamese networks based approaches have been introduced. Instead of learning a discriminative classifier online, the idea is to train (offline) a similarity function on pairs of video frames [31, 7]. Evolutions of the fully-convolutional siamese networks based approaches considerably improved tracking performance by making use of region proposals [16], and augmentating the positive samples [40]. The advantage of this kind of methods is high speed. However, the domain specific information is not used, performance of these methods is not always as good as tracking-by-detection based methods. Meanwhile, some correlation filters based methods are proposed, such as, MOSSE [1], CREST [28], MCCT [35]. However, these trackers have difficulties in obtaining a balance between the performance and speed.

Most existing methods are dependent on large scale of training datasets (e.g.,VID [25]). Hence, they can not be directly used to unsupervised drone-view vehicles.

Relation Modeling. Previous works tend to model the object relations as a post-processing by hand craft features. For example, DPM [5] modeled the relations between two objects as the co-occurrence probability by utilizing the hand craft features. Recently, Hu et al. [11] exploit the

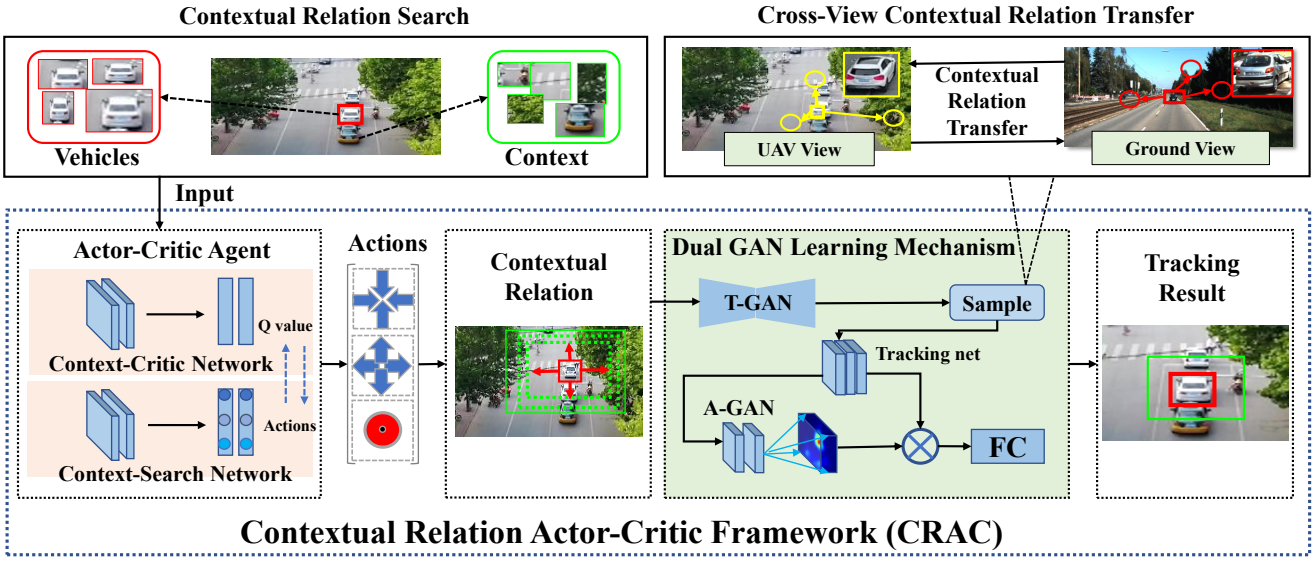

Figure 3. The architecture of our CRAC. It includes two components: (1) Contextual relation search; and (2) Cross-view contextual relation transfer. Within the actor-critic agent, the context-search network outputs coarse contextual regions for the tracking network, and the context-critic network evaluates the action set and feeds the Q value back to iteratively improve possible actions. Within the tracking network, the (improved) context is used to generate ground-view samples for UAV adaption, and A-GAN refines the critical regions via attention maps. '$\otimes$' denotes the Hadamard product.

relations during learning process. Motivated by the two works, we propose to learn the relations modeled as the co-occurrence probability in a dynamic way to facilitate the tracking performance. However, for deep learning based trackers, there is no significant improvement in object relation representation, due to the complex relations are hard to be directly modeled. Hence, we propose to learn the policy for searching the contextual relations, which does not rely on the labeled relation dataset and can be adaptively fit for the specific target. On the other hand, reinforcement learning is a principled paradigm to solve the policy learning problem in general vision tasks, and achieves remarkable success in the tasks that need interact with the environment [8, 36, 15]. Motivated by the interaction cropping method [18], which determines the boxes around the target, we propose to search the context adaptively using reinforcement learning.

Cross-View Domain Adaption. Most recent works [33, 23, 10] utilize GAN mechanism to conduct domain adaption tasks. Sankaranarayanan et al. [26] proposed to generate source-like samples with classification constraints to make the learned embedding domain adaptive. Different from the previous works that transfer the sample's appearance features or styles, our new approach tends to transfer the relation across views. Hence, we propose a dual GAN learning mechanism to firstly generate the dataset level cross-view sample in local, then refine it by an attention map in global range.

3. CRAC Framework

To transfer the cross-view appearance features, we model the relations as the view invariant contextual regions. Specifically, given an input frame I with a single target vehicle region bb_v, we define a pair-wise relation between vehicle region bb_v and its surrounding contextual region bb_c.

To use contextual relations, we need to define the relevance to evaluate how much the contextual regions could benefit the target vehicle's tracking performance. We directly evaluate the tracking score output from tracking network g for each bb_c, which shows the contextual relations with the target. The relation $\mathcal{R}$ is defined as:

$$\mathcal{R}(I) = g(bb_v, bb_c). \tag{1}$$

Here the relevance is higher, when the tracking performance is better. Furthermore, the relations are always changing even for the same target. To obtain a specific contextual relation for each vehicle target, we propose to search it with reinforcement learning framework.

To improve the unsupervised drone-view tracking performance, we propose to fully and effectively utilize the well-labeled ground-view datasets, which involves two subtasks: (1) training an actor-critic agent to find the view invariant contextual relations for providing the context clues from neighboring locations in drone view (detailed in Section 4), and (2) transferring the ground-view tracking model to adapt for the drone-view videos by generating drone-view samples with context and generating the attention map of the contextual relations (detailed in Section 5). The entire pipeline of this framework is highlighted in Fig. 3. Our pro-

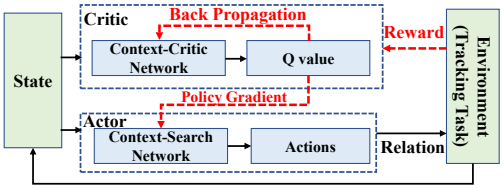

Figure 4. Actor-Critic Agent.

posed contextual relation actor-critic (CRAC) framework jointly trains the actor-critic agent and the dual GAN by maximizing the accumulated rewards from tracking task, allowing the two networks cooperatively determine the contextual relations.

4. Contextual Relation Search

We search the contextual relations with an actor-critic agent, which consists of a context-search network π and a context-critic network $\mathcal{V}$.

The actor-critic agent interacts with the environment, and takes a series of actions a_t at each step t, to optimize the context transferring policies with the reward from the tracking environment. The flow chart of the learning process is illustrated in Fig. 4. The context-search network firstly receives observations from image I in state, thereafter, it executes the sampled action a_t to dynamically search the contextual relation $\mathcal{R}$. After that, the context-critic network provides Q value according to the tracking score of the newly-generated images. The context-critic network gets the reward according to the tracking score in environment (drone-view tracking).

The basic data flow (input image and contextual relations search actions, etc.) is embedded as 'state' in this actor-critic agent. The state s_t of the actor-critic agent is represented as a tuple $s_t = (I_t, r_t, h_t)$, of which, h_t is a vector recording the history of the selected actions. The history vector h_t keeps track of the past 4 iterative actions $a_t \in h_t$. The current RGB image is $I_t \in \mathbb{R}^{w \times h \times 3}$, and the reward r_t is obtained from the tracking network.

4.1. Context-Search Network

Context-search network π aims to determine the window size of the related contextual region around the tracking target, t indicates the iteration times of the actor-critic agent. Meanwhile, the context-critic network enforces π to search the proper contextual relations benefitting the performance in drone-view tracking.

Context-Search Action. In our framework, we design 3 spatial actions $a_t \in \mathcal{A} = \{\text{enlarge, shrink, terminate}\}$. The 'enlarge' action changes the current windows size of the contextual region by a factor of 0.2, while 'shrink' changes it by 0.1. The 'terminate' action will terminate the current episode. Otherwise, the agent will continue to search more context clues until it reaches the preset maximum iteration number t_{max}. The action a_t is expected to reduce

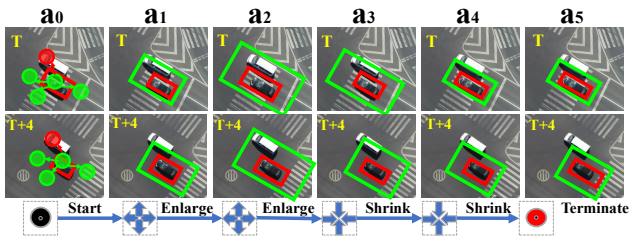

Figure 5. Example sequences of actions taken to adjust the window size. Green circles denote the related context. Red ones denote the irrelevant context.

the uncertainty of localizing the target, and allows the agent to postpone decision (i.e., after obtaining the feedback of performance from the tracking network) when the current context is ambiguous. Fig. 5 exemplifies the case that both noisy and distinct clues exist around the target in a cluttered background. The agent decides whether to enlarge the context window to obtain contextual clues or shrink to remove the noises.

Context-Search Network. Given a single image I in arbitrary view, the context-search network π should determine the actions a_t. Ideally, there should be two policy learning processes for context search in two different views. However, we assume that the contextual relation search processes in different views have the consistent searching actions. Therefore, we simplify the context search actions of both drone and ground views into a single network, the actions are separately evaluated by the context-critic network. More concretely, the context-search network π uses a Vanilla residual network [9] as the backbone network. The last layer of the context-search network is a 3-way softmax in corresponding to 3 actions.

4.2. Context-Critic Network

The context-critic network evaluates actions from the context-search network. The key components include the reward function definition, and the structure of context-critic network. We detail the two components as follows.

Reward of CRAC. The reward function is defined based on the tracking score of network $\mathcal{T}$ (value is $\mathcal{T}(;\theta_T)$). Based on Eq. 1, the function g is instantiated by $\mathcal{T}$. Hence, s_t is updated as the $\mathcal{R}_t$ dependent state $s_t = (\mathcal{R}_t, r_t, a_t)$ (which is estimated by A-GAN). We define the reward function of the CRAC as:

$$r_t = \begin{cases} \alpha \cdot sign[\mathcal{T}(s_t;\theta_T) - \mathcal{T}(s_{t+1};\theta_T)] & \text{if } a_t \neq \text{terminate}, \\ sign[\epsilon - \mathcal{T}(s_t;\theta_T] & \text{otherwise.} \end{cases} \tag{2}$$

Here the reward r_t is independent of the predicted tracking results, which indicates the gain of the tracking score at t-th time, ϵ is a threshold to discriminate whether the performance gain compared with the performance in last time is sufficient or not. The scale factor α and the threshold ϵ are set to 0.1 and 0.05 empirically. Eq. 2 indicates that the

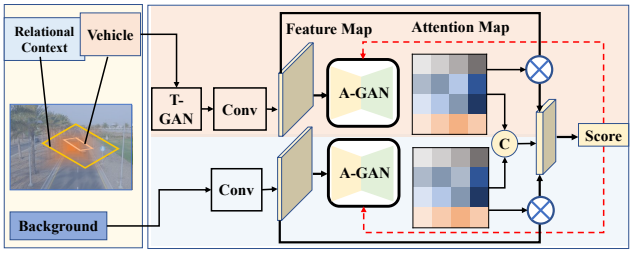

Figure 6. Dual GAN Learning Networks. 'c' is the concatenation and '$\otimes$' is the Hadamard Product.

agent receives a positive reward when the predicted action improves the tracking score from the last fully connected layer, and receives a penalty when it decreases the performance. If the agent chooses to terminate the process, the final tracking prediction must be good enough, otherwise, it would receive a large penalty. Therefore, the reward encourages the predicted contextual relations to benefit the tracking performance.

Context-Critic Network. As for the context-critic network $\mathcal{V}$ (shown in Fig. 3), we propose to approximate the Q value via a convolutional neural network, which has one scalar output approximating the Q value, with all convolution layers sharing the same structures with the context-search network π. More concretely, the context-critic network π uses a Vanilla residual network [9] as the backbone network. The last layer of the context-search network is a scalar in corresponding to Q value.

Optimization of Contextual Relation Search. We ultize a variant of asynchronous advantage actor-critic (A3C) algorithm proposed in [17] to optimize our CRAC framework. Under the framework of reinforcement learning, the goal of the contextual relation search is to maximize its expected reward over all the images. The objective function is formulated as:

$$R(a_t, s_t) = \mathbb{E}[\frac{1}{N}\sum_{i=1}^{N}\sum_{t=0}^{t_{max}}\gamma^t r_t^i(s_t, \pi(s_t;\theta))]. \quad (3)$$

Here γ is a discount factor, which controls the effect of the state in a long iteration times, r_t is the immediate reward based on the current state s_t, N is the total actions number, and t denotes the t-th iteration step, θ denotes the parameter of π. By maximizing the expected rewards, the agent learns the best policy to take actions and can explicitly balance accuracy (search for more clues in larger region) and efficiency (stop early if have a high confidence value). We use advantage function to compute the policy gradient. The output of $\mathcal{V}$ at next state s_{t+1} is the approximation of the R_{t+1} at s_{t+1} and is used to update the agent's parameters.

5. Cross-View Contextual Relation Transfer

To transfer the pre-trained model from the ground-view dataset to the drone-view dataset, we propose to transfer cross-view contextual relation with a dual GAN mechanism, including the tracking-guided GAN (T-GAN) and context attention GAN (A-GAN). The first tracking-guided GAN (T-GAN) generates the drone-view samples preserving the local discriminative features, and the second context attention GAN (A-GAN) generates the attention map to capture the global critical contextual relations. By transferring the cross-view contextual relations, we can adapt the ground-view pre-trained model to adapt for the drone-view videos. We describe the cross-view transfer schemes: contextual relation generation and contextual relation attention in details.

Contextual Relation Generation. As shown in Fig. 2, first of all, we introduce the T-GAN to transfer drone-view images, which preserves the discriminative appearance features. We aim to transfer the new samples with the following characteristics: vehicles samples look like realistic in different views; vehicles still have the discriminative feature with the background; vehicles have occlusions in drone view, e.g., the trees and buildings in drone view.

However, we lack the paired samples satisfying the above conditions. Considering huge ground-view datasets and unlabeled drone-view datasets, we generate the samples cross different views by extending the CycleGAN at the unpaired dataset level. Besides, to generate the samples satisfying the multiple objectives, we abstract the multiple objectives as a single objective that newly generated sample should improve the tracking performance. The loss of T-GAN is further defined as:

$$L(\mathcal{T}) = L_{cyc}(G_{du}, G_{ud}) + L_T(\mathcal{T}(G_{du}, G_{ud}), bb). \quad (4)$$

Here function $G_{du}(I;)$ is applied to generate the input samples from ground-view set d to drone-view set u, bb denotes the ground truth bounding box of the tracking target in the ground-view datasets (e.g., VOT), $\mathcal{T}$ denotes tracking network, L_t enforces the predicted box to be close with the ground truth. During the training (testing) phase, the loss L_T simplifies the three requirements as a unified one. Extended from CycleGAN [38], L_{cyc} tries to generate images that look similar to those in drone-view (ground-view) dataset. The two objective losses are alternatively trained, which generates realistic cross-view samples and preserves the critical appearance features related to the object tracking.

Contextual Relation Attention. We observe that the contextual relations provide global clues in distant portions of the target. In fact, the relations can be further encoded as the a more refined attention map compared with the contextual regions. To this end, we propose attention GAN (A-GAN) to estimate the attention map $\mathbf{p}$ of the contextual relations between the target and its context conditioned on ground-view relations. To enforce the attention map to capture the tracking-relevant features, we add A-GAN layers

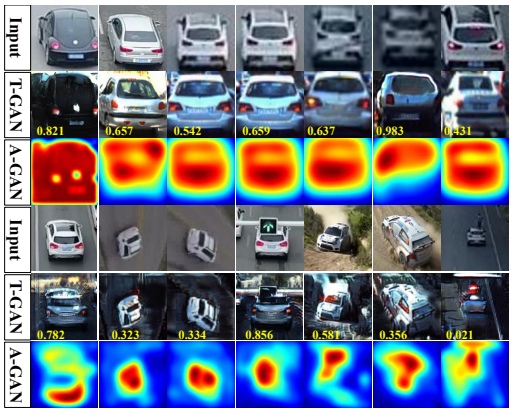

Figure 7. Samples of T-GAN and A-GAN. The scores of the generative samples are in yellow. The attention map follows the 'JET' color map.

between high level semantic feature maps (resulted from the convolution layers) and the classifier, which is in fact one branch of fully-convolution layers after the last convolutional, as shown in Fig. 6. The objective loss of the A-GAN network is defined as:

$$L_{att} = \mathbb{E}[logD(\mathbf{p} \cdot F)] + \mathbb{E}[log(1 - D(G_a(F) \cdot F))] + \lambda \mathbb{E}||G_a(F) - \mathbf{p}||. \quad (5)$$

Here the $(\cdot)$ means the Hadamard product operation on the feature maps $F \in \mathbb{R}^{M \times N \times C}$ extracted from the tracking network $\mathcal{T}$. The attention map $\mathbf{p}$ contains only one channel and has the same resolution with F. Discriminator D (i.e, classifier in $\mathcal{T}$) enforces generated attention map to be frame-specific for tracking networks. The adversarial training makes G_a (A-GAN) generate the attention maps which are less frame-specific for training D. Therefore, D will not be overfitting to the local related regions, instead, rely on more robust global features (Fig. 7). Accordingly, L_{att} provides the critical regions of the context and the target for further classification.

6. Experiments and Evaluations

We conduct extensive experiments to demonstrate the effectiveness of our CRAC framework, and compare it with state-of-the-art trackers on the vehicle subsets of Drone2018 [39], UAV123 [20], Drone Tracking Benchmark (DTB) [19] and the large scale UAV-Vehicle [4]. We further analyze the results on various challenge attributes in Drone2018 dataset (e.g., occlusions, fast motion which verifies, etc.) to verify the robustness under complex conditions. All the experiments on the four datasets are conducted in unsupervised manner.

Implementation Details. *Actor-Critic Agent:* we initialize the context-search network with a pre-trained Resnet50 model [9] and fine-tune the fully-connected layers to output context operations. The discount factor γ for the subsequent reward is set to 0.2.

Table 1. Ablation Studies Setting

Abbreviations	T-GAN	A-GAN	Agent	siamRPN	MDNET
CRAC-mdnet-A	✓	✓			✓
CRAC-mdnet-G1	✓		✓		✓
CRAC-mdnet-G2		✓	✓		✓
CRAC-mdnet	✓	✓	✓		✓
CRAC-siam-A	✓	✓		✓	
CRAC-siam-G1	✓		✓	✓	
CRAC-siam-G2		✓	✓	✓	
CRAC-siam	✓	✓	✓	✓	

Dual GAN: during the offline training phase, we use the ground-view KITTI dataset [6] and the Drone2018 datasets without tracking annotations to train the T-GAN. A-GAN is trained online with the pre-trained tracking network $\mathcal{T}$. The learning rate for training G_a and D in A-GAN are 10^{-3} and 10^{-4}, respectively.

$\mathcal{T}$ *network:* tracking network $\mathcal{T}(s_t; \theta_{\mathcal{T}})$ is extended from the network structure of MDNET [21]. We refer it as 'CRAC-mdnet'. The tracking network $\mathcal{T}$ is offline trained based on the VOT13-15 datasets [14] excluding the vehicles in drone view. With the pre-trained T-GAN, we employ the offline trained network $\mathcal{T}$ for the online tracking. During the online tracking phase, we fine-tune the the A-GAN embedded tracking network $\mathcal{T}$ with the annotations of the first frame, and conduct tracking in the subsequential frames.

To validate our CRAC framework's generalization ability on different $\mathcal{T}$ networks, we utilize the SiamRPN [31] as the $\mathcal{T}$ network. The pre-trained network is trained on VOT13-15 datasets [14], we refer it as 'CRAC-siam'.

Evaluation Metrics. To fairly compare with the previous works, our backbone feature extractor is based on the first three convolutional layers from the VGGM model [27], as that in [29]. Besides, our CRAC is implemented using the Pytorch library and MatConvNet toolbox [32]. We follow the standard evaluation approaches. In the Drone2018, UAV123, DTB, and UAV-Vehicle datasets, we use the one-pass evaluation (OPE) with 'precision' and 'success plots' metrics. The precision metric measures the frame locations rate, that is within a certain threshold distance w.r.t ground truth locations. The error threshold of 20 pixels is used for ranking. The success plot metric is set to measure the overlap ratio between the predicted bounding boxes and the ground truth.

Ablation Studies. To verify the contribution of each component in our method, we evaluate several configurations of our approach. The effectiveness of four key elements of our CRAC framework is evaluated on Drone2018 dataset: including $\mathcal{T}$ network (MDNET or SiamRPN), T-GAN, A-GAN, Actor-Critic Agent (agent), which are described in Tab. 1. The performance is shown in Fig. 11.

The performances of all the incomplete configurations are not as good as the full configuration of our CRAC-mdnet (CRAC-siam), and each component in our tracking algorithm is helpful to improve performance. The learned attention map contributes most, which improves CRAC-

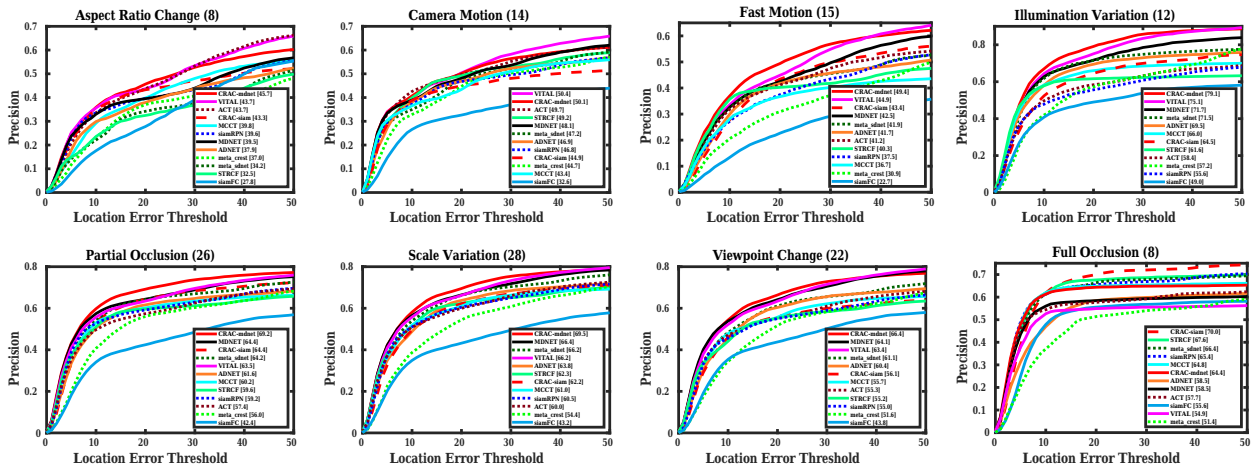

Figure 8. Comparison over the challenging cases for the drone-view tracking, including background clutter, aspect ration change, camera motion, etc.

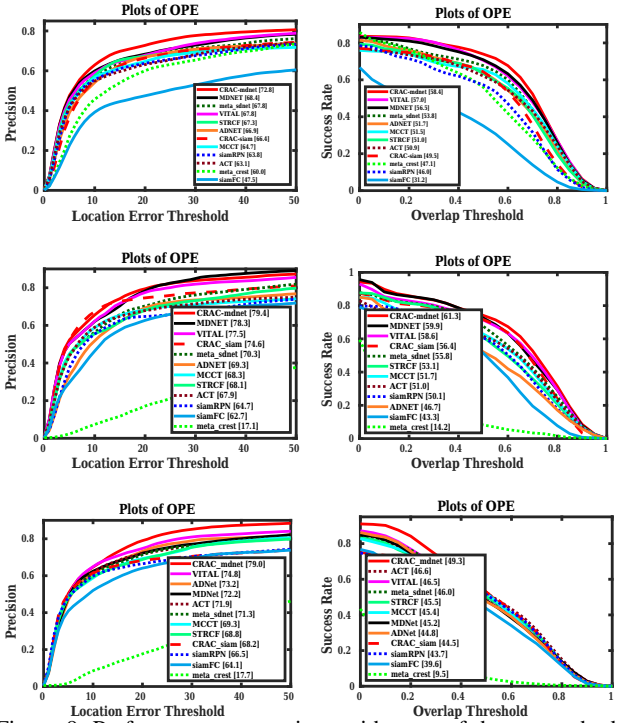

Figure 9. Performance comparison with state-of-the-art methods on Drone2018 (1st row), merged UAV123, DTB (2nd row) and UAV-Vehicle (3rd row) dataset.

mdnet from 66.7 to 72.8 in precision plots. This is because that attention map tends to flexibly leverage more reliable clues provided by contextual relations. It can be concluded that, the relations refined by attention map from our CRAC framework can benefit the tracking task in drone view.

Further, CRAC-mdnet and CRAC-siam networks improve the MDNET and siamRPN networks by 4.4 and 2.6 in precision plots, which verifies the generalization ability of different $\mathcal{T}$ networks.

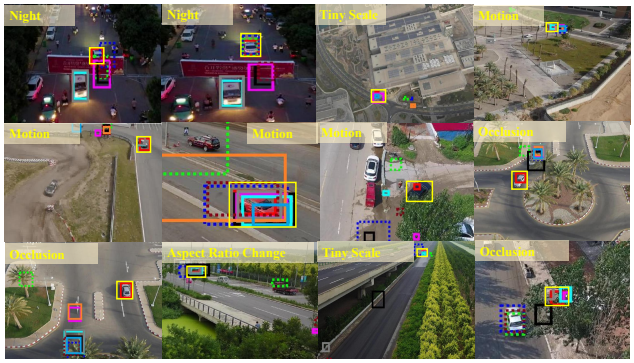

Figure 10. Analysis of the performance under occlusions, motion, tiny scale targets, etc. The results demonstrate our CRAC is robust to these hard cases. (Same legend with the Fig. 9, yellow box denotes ground truth.)

We also evaluate the convergence performance of our CRAC in different iteration times: 20, 50, 80, 100 (denoted as CRAC2, CRAC5, CRAC8 and CRAC10), the performance is stable and the 80-iteration case performs best, which is denoted as CRAC-mdnet.

Besides, to evaluate the contribution of training on KITTI dataset, we also conduct tracking in the vehicle subset of KITTI with the MDNET (SiamRPN) offline training, denoted as CRAC-siam-offline (CRAC-mdnet-offline), which demonstrates that the trackers trained on ground-view dataset will not perform well when directly used to drone-view, and it is significant that CRAC can transfer the contextual relations across views.

Comparison with State-of-the-Art Methods. We compare our CRAC-mdnet (CRAC-siam) with 10 state-of-the-art trackers, including MDNET [21], VITAL [29], ACT [2], ADNET [37], STRCF [18], meta-crest and meta-sdnet [22], SiamFC [31], SiamRPN [31], and MCCT [35]. We evaluate all the trackers on 122 testing video sequences from the drone-view datasets with distance precision and overlap

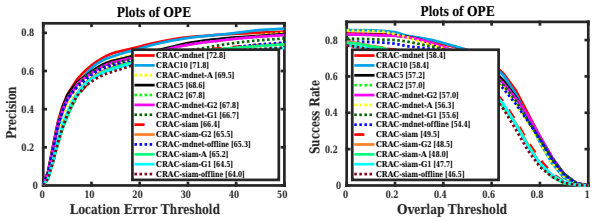

Figure 11. Ablations studies on Drone2018 dataset, setting is in Tab. 1.

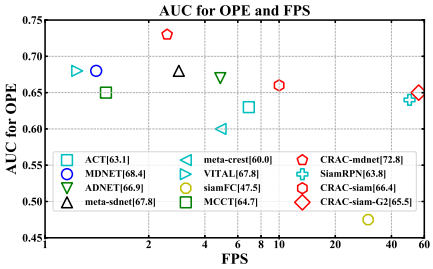

Figure 12. Performance and speed of our trackers and some state-of-the-art trackers on Drone2018 dataset. More closed to 'top' means higher precision, and more closed to 'right' means faster. CRAC-siam-G2 ranks top 3 in EAO while keeps a high speed at 56 FPS.

success metrics. Fig. 9 shows that our CRAC model outperforms other methods in precision and success plots on Drone2018 (50 sequences), DTB (7 sequences), UAV123 (17 sequences), UAV-Vehicle (48 sequences, S1302 and S1310 are excluded) datasets. We merge the vehicle sequences in DTB and UAV123 to simplify the performance comparison.

Compared to the representative tracking-by-detection trackers (MDNET and VITAL), we attribute our performance improvement to the generated drone-view contextual relations, which facilitates training robust classifiers. On most of the never-before-seen sequences, other trackers fail to locate the target objects or estimate the scale incorrectly. Our CRAC-mdnet (CRAC-siam) improves the performance, which bridges the gap between sample generation and online (offline) tracking.

Compared with the state-of-art correlation filter based trackers (e.g., meta-crest, MCCT), our CRAC-mdnet (CRAC-siam) tracker emphasizes the most robust features provided by the contextual relations and performs better.

The reinforcement learning scheme makes bounding box search be adaptive with the features. Compared with the previous reinforcement learning methods, such as ACT [2], ADNET [37], our CRAC-mdnet (CRAC-siam) can transfer the target-related contextual relations. Therefore, our CRAC-mdnet (CRAC-siam) outperforms the second and third high-performance works by a large margin on all of the four datasets, e.g., on the Drone2018 dataset, CRAC-mdnet (CRAC-siam) respectively gains 72.8-70.2 (66.4-65.6) and 72.8-69.7 (66.4-48.9) precision improvement w.r.t the two high-performance trackers.

Besides, our CRAC-siam network has a high speed compared with the corresponding baseline trackers. The performance and speed are shown in Fig. 12

Evaluations Under Challenging Conditions. Drone-view vehicle tracking may confront with challenging conditions, such as the sudden high-speed camera motion, partial/full occlusion, background clutter, etc. We quantitatively evaluate the robustness of our CRAC-mdnet (CRAC-siam) on 8 kinds of challenging cases with other state-of-the-art methods. The results are shown in Fig. 8. Our CRAC-mdnet (CRAC-siam) can well handle the large appearance variations caused by aspect ratio change (deformation), partial occlusion, view point change, etc. The difficulties in these conditions can be relieved by transferring information from the ground-view datasets. It can be concluded that, our CRAC-mdnet method benefits the feature extraction under the illumination variation condition, gaining performance improvement from 71.7 (MDNET) to 79.1 in precision plots, and it also gives rise to a large improvement in the occlusion and camera motion conditions. Fig. 10 shows some challenging cases improved by CRAC. When the cameras have drift problem, sudden appearance change makes other trackers lose the target completely in a jump sequence, but our CRAC-mdnet (CRAC-siam) can still track the target steadily. However, under the full occlusion condition, both CRAC-mdnet and MDNET can not perform as well as other cases. It is probable that their online updating strategy will easily lose the original target when the target is fully occluded. More examples are provided in our supplementary material.

7. Conclusions and Future Works

In this paper, we propose a novel CRAC framework by transferring the contextual relations from ground-view to drone-view scenes, which can reduce the dependence on large-scale well-labeled dataset. The contextual relations are dynamically modeled as the relevant contextual regions via an actor-critic agent, which can adaptively leverage the contextual relations in tracking tasks, and enables the tracking network to focus on the critical regions surrounding the tracking target. Meanwhile, we propose the dual GAN learning mechanism to transmit the relations. Extensive experiments on various benchmarks demonstrate that our CRAC outperforms state-of-the-art trackers.

When the vehicles are fully sheltered from occlusions in a long-term sequence, and there is a similar target in the contextual regions, our CRAC will enforce the tracker to locate on the wrong target. In future, we will add temporal constraints to improve the performance. Besides, we will apply our CRAC to more cross-view transferring tasks, such as vehicle Re-ID and segmentation, etc.

References

[1] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui. Visual object tracking using adaptive correlation filters. In *CVPR*, pages 2544–2550, June 2010.

[2] B. Chen, D. Wang, P. Li, S. Wang, and H. Lu. Real-time 'actor-critic' tracking. In *ECCV*, pages 318–334, September 2018.

[3] X. Dong, J. Shen, W. Wang, Y. Liu, L. Shao, and F. Porikli. Hyperparameter optimization for tracking with continuous deep q-learning. In *CVPR*, pages 518–527, June 2018.

[4] D. Du, Y. Qi, H. Yu, Y. Yang, K. Duan, G. Li, W. Zhang, Q. Huang, and Q. Tian. The unmanned aerial vehicle benchmark: object detection and tracking. In *ECCV*, pages 370–386, September 2018.

[5] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 32(9):1627–1645, 2010.

[6] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 32(11):1231–1237, 2013.

[7] Q. Guo, W. Feng, C. Zhou, R. Huang, L. Wan, and S. Wang. Learning dynamic siamese network for visual object tracking. In *ICCV*, pages 1763–1771, October 2017.

[8] J. Han, L. Yang, D. Zhang, X. Chang, and X. Liang. Reinforcement cutting-agent learning for video object segmentation. In *CVPR*, pages 9080–9089, June 2018.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, June 2016.

[10] W. Hong, Z. Wang, M. Yang, and J. Yuan. Conditional generative adversarial network for structured domain adaptation. In *CVPR*, pages 1335–1344, June 2018.

[11] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei. Relation networks for object detection. In *CVPR*, pages 3588–3597, June 2018.

[12] C. Huang, S. Lucey, and D. Ramanan. Learning policies for adaptive tracking with deep feature cascades. In *ICCV*, pages 105–114, October 2017.

[13] H. Kiani Galoogahi, A. Fagg, and S. Lucey. Learning background-aware correlation filters for visual tracking. In *ICCV*, pages 1135–1143, October 2017.

[14] M. Kristan, J. Matas, A. Leonardis, T. Vojir, R. Pflugfelder, G. Fernandez, G. Nebehay, F. Porikli, and L. Čehovin. A novel performance evaluation methodology for single-target trackers. *TPAMI*, 38(11):2137–2155, 2016.

[15] S. Lan, R. Panda, Q. Zhu, and A. K. Roy-Chowdhury. Ffnet: Video fast-forwarding via reinforcement learning. In *CVPR*, pages 6771–6780, June 2018.

[16] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu. High performance visual tracking with siamese region proposal network. In *CVPR*, pages 8971–8980, July 2018.

[17] D. Li, H. Wu, J. Zhang, and K. Huang. A2-rl: Aesthetics aware reinforcement learning for image cropping. In *CVPR*, pages 8193–8201, June 2018.

[18] F. Li, C. Tian, W. Zuo, L. Zhang, and M.-H. Yang. Learning spatial-temporal regularized correlation filters for visual tracking. In *CVPR*, pages 4904–4913, June 2018.

[19] S. Li and D.-Y. Yeung. Visual object tracking for unmanned aerial vehicles: A benchmark and new motion models. In *AAAI*, pages 4140–4146, February 2017.

[20] M. Mueller, N. Smith, and B. Ghanem. A benchmark and simulator for uav tracking. In *ECCV*, pages 445–461, October 2016.

[21] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*, pages 4293–4302, June 2016.

[22] E. Park and A. C. Berg. Meta-tracker: Fast and robust online adaptation for visual object trackers. In *ECCV*, pages 569–585, September 2018.

[23] K. Regmi and A. Borji. Cross-view image synthesis using conditional gans. In *CVPR*, pages 3501–3510, June 2018.

[24] L. Ren, X. Yuan, J. Lu, M. Yang, and J. Zhou. Deep reinforcement learning with iterative shift for visual tracking. In *ECCV*, pages 684–700, September 2018.

[25] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015.

[26] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *CVPR*, pages 8503–8512, June 2018.

[27] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[28] Y. Song, C. Ma, L. Gong, J. Zhang, R. Lau, and M.-H. Yang. Crest: Convolutional residual learning for visual tracking. In *ICCV*, pages 2555–2564, October 2017.

[29] Y. Song, C. Ma, X. Wu, L. Gong, L. Bao, W. Zuo, C. Shen, R. W. Lau, and M.-H. Yang. Vital: Visual tracking via adversarial learning. In *CVPR*, pages 8990–8999, June 2018.

[30] J. Supancic, III and D. Ramanan. Tracking as online decision-making: Learning a policy from streaming videos with reinforcement learning. In *ICCV*, pages 322–331, October 2017.

[31] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. S. Torr. End-to-end representation learning for correlation filter based tracking. In *CVPR*, pages 2805–2813, July 2017.

[32] A. Vedaldi and K. Lenc. Matconvnet – convolutional neural networks for matlab. In *ACMMM*, pages 689–692, October 2015.

[33] R. Volpi, P. Morerio, S. Savarese, and V. Murino. Adversarial feature augmentation for unsupervised domain adaptation. In *CVPR*, pages 5495–5504, June 2018.

[34] L. Wang, W. Ouyang, X. Wang, and H. Lu. Stct: Sequentially training convolutional networks for visual tracking. In *CVPR*, pages 1373–1381, June 2016.

[35] N. Wang, W. Zhou, Q. Tian, R. Hong, M. Wang, and H. Li. Multi-cue correlation filters for robust visual tracking. In *CVPR*, pages 4844–4853, June 2018.

[36] X. Wang, W. Chen, J. Wu, Y.-F. Wang, and W. Y. Wang. Video captioning via hierarchical reinforcement learning. In *CVPR*, pages 4213–4222, June 2018.

[37] S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Young Choi. Action-decision networks for visual tracking with deep reinforcement learning. In *CVPR*, pages 2711–2720, July 2017.

[38] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, pages 2223–2232, October 2017.

[39] P. Zhu, L. Wen, X. Bian, L. Haibin, and Q. Hu. Vision meets drones: A challenge. *arXiv preprint arXiv:1804.07437*, 2018.

[40] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu. Distractor-aware siamese networks for visual object tracking. In *ECCV*, pages 101–117, September 2018.