Object Tracking in Satellite Videos Based on Convolutional Regression Network With Appearance and Motion Features

Zhaopeng Hu, Daiqin Yang, *Member, IEEE*, Kao Zhang, and Zhenzhong Chen, *Senior Member, IEEE*

***Abstract*—Object tracking is one of the most important components in numerous applications of computer vision. Remote sensing videos provided by commercial satellites make it possible to extend this topic into the earth observation domain. In satellite videos, typical moving targets like vehicles and planes only cover a small area of pixels, and they could easily be confused with surrounding complex ground scenes. Similar objects nearby in satellite videos can hardly be differed by appearance details due to the resolution constraint. Thus, tracking drift caused by distractions is also a thorny problem. Facing challenges, traditional tracking methods such as correlation filters with hand-crafted visual features achieve unsatisfactory results in satellite videos. Methods based on deep neural networks have demonstrated their superiority in various ordinary visual tracking benchmarks, but their results on satellite videos remain unexplored. In this article, deep learning technologies are applied to object tracking in satellite videos for better performance. A simple regression network is used to combine a regression model with convolutional layers and a gradient descent algorithm. The regression network fully exploits the abundant background context to learn a robust tracker. Instead of hand-crafted features, both appearance features and motion features, which are extracted by pretrained deep neural networks, are used for accurate object tracking. In cases when the tracker encounters ambiguous appearance information, the motion features could provide complementary and discriminative information to improve tracking performances. Experimental results on various satellite videos show that the proposed method achieves better tracking performance than other state-of-the-arts.**

***Index Terms*—Convolutional neural networks (CNNs), deep learning, object tracking, satellite video.**

I. Introduction

VISUAL object tracking is a fundamental task in computer vision with various practical applications. Given the initialized state (e.g., position and size) of a target object in a frame of a video, the goal of tracking is to estimate the states of the target in the following frames [1]. With the advancement in remote sensing satellite platforms, e.g., Skysat-1 and Jilin-1 satellite, high-resolution videos gazing a specific area of the ground are available. These videos bring about new means for monitoring land surface. Accordingly, satellite video object tracking becomes an emerging research topic. It can be used for many important applications, such as estimating traffic density [2], monitoring sea ice, and fighting wildfires [3].

In recent years, commercial satellite technology has achieved great progress in capturing high-resolution videos. Among the video satellites, Skysat-1 launched by Skybox Imaging Company in 2013 is the first one in the globe to capture panchromatic high-resolution videos [4]. The resolution of Skysat-1 reaches meter level, and the ground coverage of its image products is about 2 km^2. The Jilin-1 Smart Video Satellites produced by China-Changguang Satellite Technology Company, Ltd. provides 4K high-definition imagery in real-time monitoring and can capture live imagery for faster response. The early video products, which cover 4.6 × 3.4 km ground area, have a resolution of about 1.13 m, and the newest products could reach less than 0.92-m resolution.

Compared with ordinary videos, different characteristics of satellite videos need to be taken into consideration when tracking the moving objects. Since satellite videos are collected in space, common challenges in ordinary tracking datasets such as occlusion and scale changing are not severe. However, compared with the large size of the whole frame, the moving objects usually cover only a small area of pixels, providing little feature and texture information. Owing to this attribute, different targets in the same category could have similar appearances, which could mislead the tracker to error results. Besides, the targets sometimes could also be mixed into the land background when passing through shadow areas due to limited contrast in between. This could cause model drift.

Derived from the pioneering work of MOSSE [5] and CSK [6], the trackers based on a discriminative correlation filter (DCF) [7]–[10] have taken the leading position among tracking algorithms in recent years. By the dense sampling strategy and transforming operations to the Fourier domain, DCF trackers perform well in terms of both accuracy and speed. However, these methods face challenges when applied to satellite videos, which have a small target size and complex background [11]. With the significant success of the convolutional neural network (CNN) models in many vision tasks, researchers are also inspired to explore the capability of deep learning for tracking problems. State-of-the-art CNN-based trackers [12]–[15] have made great progress toward this goal. Trackers based on deep learning are

Manuscript received November 3, 2019; revised January 15, 2020; accepted January 31, 2020. Date of publication February 11, 2020; date of current version February 27, 2020. This work was supported by the National Key R&D Program of China under Contract 2018YFB0505501 and Contract 2018YFB0505500. *(Corresponding author: Daiqin Yang.)*

The authors are with the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China (e-mail: cyperhzp@whu.edu.cn; dqyang@whu.edu.cn; zhangkao@whu.edu.cn; zzchen@whu.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2020.2971657

more robust compared with traditional methods. But they still need to be adjusted for satellite videos.

There have been several studies aiming to solve the problem of object tracking in satellite videos [11], [16]–[19]. Most of these trackers use hand-crafted feature representations, such as invariant moment, histogram of oriented gradient (HOG), or optical flow. Deep neural network features have been successfully transferred to various computer vision tasks. Activations from convolutional layers of the CNN are semantically meaningful and contain structural information crucial for the localization task. These motivate us to harness deep features for the tracking task on satellite videos. A convolutional regression network (CRN) [20], [21] is adopted as the backbone of our tracker. The CRN uses convolutional layers to solve the regression problem with gradient descent in an end-to-end learning manner.

To solve the tracking drift problem caused by similar targets nearby, appearance features and motion features [22] are combined together to improve robustness. High-level motion features, which are extracted based on the optical flow values, provide complementary representations beyond appearance features. Targets with similar appearance could be discriminated by differences in motion features. This can alleviate the tracking drift problem. In our proposed method, a convolutional regression network with appearance and motion features (CRAM), two regression networks are trained with different appearance and motion features, respectively, and their responses are then integrated for final target location prediction. The contributions of each response will be weighted by their qualities, which are measured by the peak-to-sidelobe ratio (PSR) [5].

The main contributions of this article are summarized as follows.

1) A CRAM tracker is designed for object tracking in satellite videos based on the CRN, which replaces the commonly used DCF framework. The regression network fully exploits the abundant background context with end-to-end learning, increasing tracker's discriminative power during the tracking process.
2) Appearance and motion features extracted by deep neural networks are combined together in use. The tracking performance could be improved benefiting from the high-level feature representations and complementary information.

The rest of this article is organized as follows. Section II gives an overview of the prior studies relevant to this article. The details of the proposed tracking method are presented in Section III. The experimental results on our satellite video dataset are presented in Section IV. Finally, Section V concludes this article.

II. Related Work

A. DCF-Based Object Tracking

Trackers originated from the DCF framework have been the recent trends in object tracking. The key procedure is to solve the regression problem by directly learning a mapping from regularly dense samples of target objects to soft Gaussian labels for target positions estimation. Formally, the goal is to learn a linear function $y = \theta^T \cdot x$, where x denotes the m-dimensional feature of a sample pixel and y is the corresponding label of the pixel. The regression coefficients θ can be learned by solving the following minimization problem:

$$\arg\min_{\theta} \ \|X\theta - Y\|^2 + \lambda\|\theta\|^2 \tag{1}$$

where matrix $X \in \mathbb{R}^{m\times n}$ consists of n feature samples, representing the features extracted from the search area, while $Y \in \mathbb{R}^n$ represents the labels of these n pixels in the search area. λ is the regularization parameter to penalize overfitting. This problem has a closed-form solution, which is given by

$$\theta = (X^T X + \lambda I)^{-1} X^T Y \tag{2}$$

where I is an identity matrix.

The direct solution of (2) becomes computationally prohibitive when m and n are large. The CSK [6] tracker introduces circularly shifting of samples and kernel trick to form a correlation filter for fast computation, and this strategy is adopted by other DCF trackers. Benefit from the structure of the DCF framework, multichannel features could be incorporated to improve tracking performance. KCF [7] adopts HOG feature to increase accuracy, and DSST [8] further exploits on solving the scale estimation problem. However, the extra background information needed for training samples generation and the boundary effects induced by the circulant structure impose limitation to the performance of the regression model.

B. Deep CNN Used in Object Tracking in General Scenes

With the emergence of deep CNN in computer vision field, given architectures such as AlexNet [23] and VGGNet [24] trained on the ImageNet dataset [25] are studied in object tracking [12], [15], and [26] as feature extractors. Multiple convolutional layers are employed in hierarchical or joint ways to integrate different visual information. The CF2 [12] tracker adaptively learns correlation filters on different convolutional layers to encode the target appearance and hierarchically infer the maximum response of each layer to locate targets. In ECO [15], deep CNN layers along with handcrafted features are integrated with a factorized convolution operator. Deeper and more sophisticated networks boost the performance of the state of the arts. However, the increasing translation invariance in deeper layers resulting from spatial pooling operations hinders precise localization in DCF trackers. Considering the low resolution of satellite videos, layers used for the deep CNN model need to be carefully selected.

The object tracking community later started training network architectures to take full advantage of the benefits of end-to-end learning [13], [14], [27]. In MDNet [13], a multidomain network is proposed to learn the shared representation of targets from multiple annotated video sequences for tracking. The fully convolutional siamese network is used in [14] to directly learn strong embeddings from large amounts of video data. The offline training design makes this tracker operate beyond real-time with competitive performance. SiamRPN [27] proposed in 2018 combines a region proposal network [28] with the siamese network to get a more accurate bounding box. It got top performance with real-time speed and could perform better with more training data. However, due to the scarcity of annotated datasets of satellite videos, it is difficult to develop an end-to-end deep learning tracker for satellite platforms.

Different from trackers mentioned above, deep regression trackers apply CNNs to solve the regression problem in tracking. The FCNT [29] tracker makes the first effort to learn regression networks over two CNN layers. The output response maps from different layers are switched according to their confidence to locate target objects. Extended from FCNT, ensemble learning is exploited in STCT [30] to select CNN feature channels. The studies in [21], [31], and [32] all construct their trackers based on one convolutional layer. The kernel size of the layer is deduced from the regression model, rather than empirically set. One single-convolutional-layer-based regression model is less likely to be overfitted and more efficient to compute. DSLT [20] further studies the bottleneck in training the regression network. It uses shrinkage loss to penalize easy training data and applies residual connections to exploit multilevel semantic abstraction. In our work, we follow the one-layer CRN model to construct our tracker, with both appearance and motion features extracted by pretrained deep CNNs.

C. Satellite Video Object Tracking

Researchers have developed several methods for single-target tracking in satellite videos in recent years. By viewing the moving vehicles as point targets, Wu *et al.* [16] introduce a method based on the Bayesian classification with the grayscale feature and the motion smoothness constraint. Du *et al.* [11] use a three-frame difference method in combination with a correlation filter for satellite video tracking. A specific strategy is proposed, taking advantage of the KCF [7] tracker and the three-frame difference algorithm, to build a strong tracker. Guo *et al.* [19] proposed a DCF-based high-speed tracker, which applies a Kalman filter to correct the tracking trajectory. In these studies, handcrafted appearance features like invariant moment and HOG are not capable enough for robust tracking, and it is indispensable for the combination of motion detection algorithms such as three-frame difference or Kalman filter.

Recently, Du *et al.* [18] have proposed a multiframe optical flow tracker (MOFT) to further improve the tracking performance on satellite videos. Shao *et al.* [3], [17] employ the velocity feature and the inertia mechanism to construct a velocity correlation filter (VCF). Unlike ordinary videos, in which well-designed appearance features are capable of distinguishing similar objects in the same category, such as different cars or persons, in satellite videos, appearance features are not enough due to the resolution constraint. This is the reason why MOFT and VCF use features derived from optical flow, which are rarely used in ordinary visual tracking tasks. However, these studies neglect the combination of appearance features, and the results can be further improved.

Inspired by the prior works, our proposed method CRAM fuses the motion features with standard appearance features to exploit the complementary information. Two regression networks are constructed with two different features, and their results are then integrated together. This strategy could improve the robustness of the tracker in the case of challenging scenes in satellite videos.

III. Proposed Method

The network architecture of the CRAM includes a feature extraction part and a convolutional regression part. In the convolutional regression part, there are two CNN modules, named *Net-A* and *Net-M*, forming two pathways in the execution of the CRAM. They are designed with the same network structure but trained with different features. As shown in Fig. 1, first, a search area centered at the target position in the previous frame will be cropped and upsampled. Next, appearance features and motion features are extracted separately from the RGB image and optical flow image by the feature extraction network and fed to two separate CRNs. The output response maps of *Net-A* and *Net-M* will then be combined together according to their PSR. Finally, the predicted target position is set to the position, where the highest response value appears. During tracking, the regression model is regularly updated to adapt to current object and background appearance. Details about the CRAM are depicted below.

A. Feature Extraction

1) Deep Appearance Features: After training for a particular vision task such as image classification, the feature representations learned by CNNs have proven to be generic and can be used for a variety other vision problems. The deep features are usually more discriminative than handcrafted features and possess high-level visual information. Structural information contained in the convolutional features can also be used for the localization purpose. Since it is nontrivial to train a deep feature extraction network for satellite images from scratch, the CRAM adopts the VGG16 [24] model, which is pretrained on the ImageNet dataset. The VGG16 network contains five blocks with 13 convolutional layers in total. The CRAM uses the first three convolution blocks and takes the activations from the seventh convolutional layer (conv3-3), after the rectified linear unit operation. Only the first two max-pooling layers are retained, so the features will have a spatial stride of four pixels, compared to the input image patch, and 256 feature channels.

It should be noted that the response map obtained in later regression calculation has the same spatial size with input features, namely one-fourth the size of the original search area. Given hundreds or thousands of pixels of the target area, the impact of this size shrinkage is neglectable for ordinary visual tracking task. However, in satellite videos, with tens of target pixels, this size shrinkage will bring significant performance degradation (see Section IV-B2 for more details). Therefore, in the CRAM, the search areas will be upsampled four times before feature extraction, in order to make the spatial size of the features and response map the same as the original input patch.

2) Deep Motion Features: Optical flow algorithms calculate the motion of each pixel between two frames, and the obtained optical flow is useful in various applications, such as action recognition and object tracking. Instead of directly using the

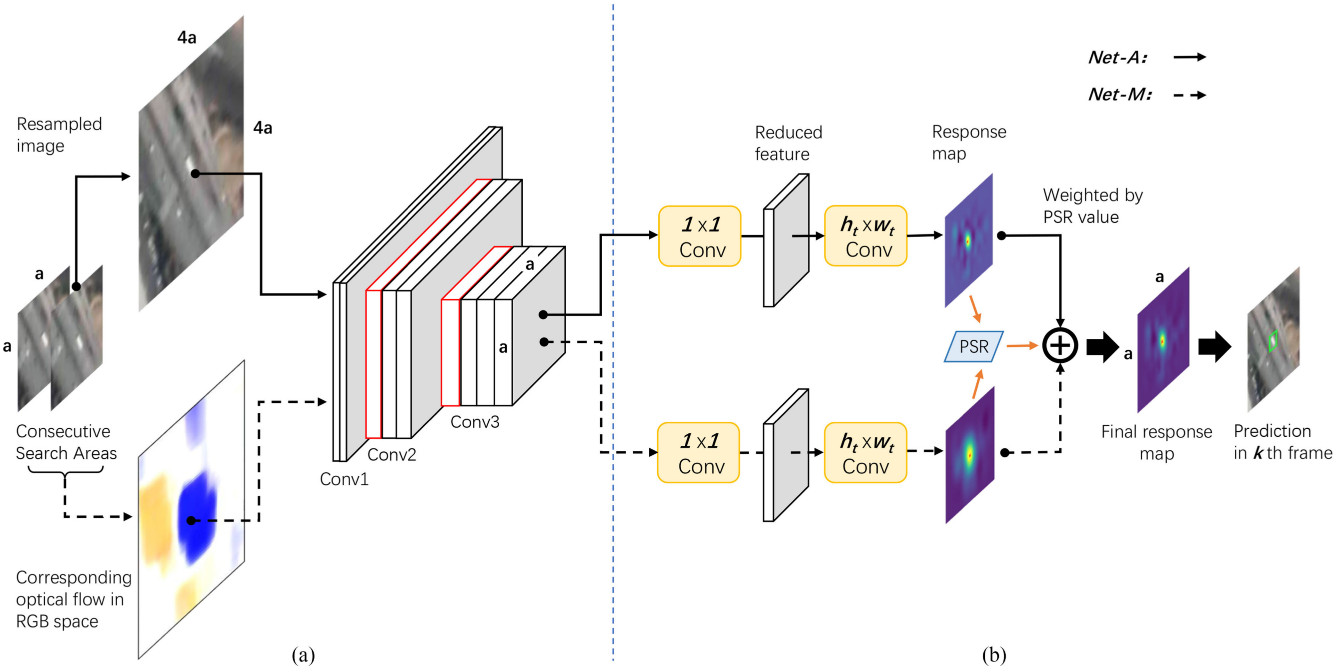


Fig. 1. Overview of the proposed tracking method CRAM. (a) Feature extraction network. (b) CRN-based tracker, including *Net-A* and *Net-M*, in which appearance and motion features are used, respectively.

original optical flow feature like MOFT and VCF, we adopt a similar strategy as in [22], where high-level deep motion features are extracted from optical flow fields by a pretrained CNN. In this way, objects with different motion characteristics will exhibit different feature representations. This could help the tracker in distinguishing the target under tracking from other similar distractors of the complex background.

Similar as appearance feature extraction, the CRAM reuses the weights and biases of the first seven convolutional layers in the VGG16 network. The optical flow estimator comes from [33]. Before deep motion feature extraction, each pixel's 2-D optical flow vector should be transformed into a pseudo-3-D RGB space. The CRAM adopts the idea provided by Baker *et al.* [34], in which Hue, Saturation, Value (HSV) color transformation is used. To use HSV transformation, direction and magnitude of each optical flow vector are taken as Hue and Saturation, respectively. And the Value plane is set to a constant. As shown in Fig. 1, after the HSV to RGB transformation of each pixel, the optical flow image is transformed to an RGB image for feature extraction.

B. Convolutional Regression Network

Instead of directly calculating the closed-form solution as (2), the regression model can be reformulated as a one-layer convolution operation, in order to regress the dense sampling of inputs to soft labels. The translation-invariant structure of a convolutional layer makes it a helpful tool to compute the inference of a large-scale linear regression model. Suppose that the training samples are cropped from the search area in a dense sliding-window manner with the spatial size of $h_t \times w_t$, where h_t and w_t are the height and width of object. Then, the inference of the linear regression can be computed by forward-propagating the search patch through a single convolutional layer with kernel size of $h_t \times w_t$. The regression coefficients θ are now represented by weights w and bias b of the convolutional layer, which are trained by training samples with gradient descent. The target position is estimated by searching for the location with the maximum value of the output response map. Training sample patches obtained by the sliding-window operation avoid negative effects caused by unreliable artificial sample patches used in the DCF. The search area could be arbitrarily enlarged without worrying about boundary effects, and the background information can also be fully exploited.

C. Tracking Framework

1) Regression Model Initialization: The regression models of both *Net-A* and *Net-M* should be trained before usage. To remove redundant layers of deep features, a 1×1 convolutional layer is inserted in front of the regression layer, cutting down the feature channels to 32. For *Net-A*, the search area image in the first frame surrounding the given target and its corresponding response map are used for training. The training response map is generated by a 2-D Gaussian distribution, where the maximum value coincides at the center of the tracking target. With low image resolution and complex background changing, the calculated optical flows often contain unneglectable noises, which could contaminate the regression model during training. Therefore, for *Net-M*, it is designed to be trained with a batch of optical flow images collected in five consequent frames. The response map is set to the foreground mask instead of the Gaussian distribution. The predicted target location by *Net-A* will be used as ground truth of the foreground mask, assuming the shape of the target unchanged. Labels in the mask are marked either one or zero, denoting object or background. After training, *Net-M* will be qualified to predict the foreground map through deep motion features extracted from optical flows. During training, all the parameters in the CRN layers are randomly initialized following zero-mean Gaussian distribution.

2) Online Detection: The prediction of the target location is formulated as the mapping of the peak value location in the response map to the image coordinate. When there exists distractors that have similar appearances as the tracking target in the search window, the response map generated by *Net-A* will have more than one peaks. In this case, the prediction based only on appearance features is vulnerable. And the regression result of *Net-M*, which employs motion features, can be a crucial complementary. By combining deep appearance and motion features, the complementary information can provide robust tracking. Specifically, the position (x_t, y_t) can be optimized with the following objective function:

$$\underset{x_t, y_t}{\arg\max}\ \alpha A(x_t, y_t) + (1-\alpha) F(x_t, y_t) \tag{3}$$

where

$$F(x_t, y_t) = \frac{1}{w_t h_t} \sum_{(i,j)\in R(x_t, y_t, w_t, h_t)} M(i,j). \tag{4}$$

A and M denote the response maps of *Net-A* and *Net-M*, respectively. $F(x_t, y_t)$ is the local average of M around target position (x_t, y_t); $R(x_t, y_t, w_t, h_t)$ is the area defined by the bounding box with size w_t, h_t at x_t, y_t. Equation (4) tries to turn the predicted foreground mask to a probability map. α is a weighting factor to balance the contributions of the two regression results.

The value of weighting factor α is not fixed during tracking. When the prediction by *Net-A* is more reliable, the weight of its response is bigger and *vise versa*. The PSR is adopted as a measurement of the tracking confidence of *Net-A* and *Net-M*. To compute the PSR value, the response map is split into the peak area and the sidelobe. The peak area is covered by a square window centered at the position with the maximum response (see Fig. 2), and the sidelobe is the rest area. The size of the peak area is set to one-fourth the size of the response map. The PSR is defined as

$$\text{PSR} = \frac{g_{\max} - \mu_{\text{sl}}}{\sigma_{\text{sl}}} \tag{5}$$

where $g_{\max}$ is the maximum response value and μ_{sl} and σ_{sl} are the mean and standard deviation of the sidelobe, respectively. This value shows the peak strength of the response map. When tracking results are poor, there could be multiple ambiguous subpeak areas in the response map. These subpeak areas will increase both μ_{sl} and σ_{sl} and degrade the PSR value. Denoting the PSR values of *Net-A* and *Net-M* as PSR_A and PSR_M, α is calculated as

$$\alpha = \frac{\text{PSR}_A}{\text{PSR}_A + \text{PSR}_M}. \tag{6}$$

3) Model Update: With changing appearance and motion features, the regression networks should be regularly updated. In satellite videos, given the much larger field of view, the changing speed of the tracking target is relatively slow. So, the updating frequency of the regression models is set to a moderate value of every five frames in the CRAM. For each updating frame, new training patches and label maps are generated from the latest prediction results, and the regression networks are fine-tuned for a few iterations. To alleviate noisy updates, the regression model will only be updated if the maximum value of response map surpasses a threshold th_update. In other words, the maximum value of the response map will be checked every five frames to decide whether a model updating is needed.

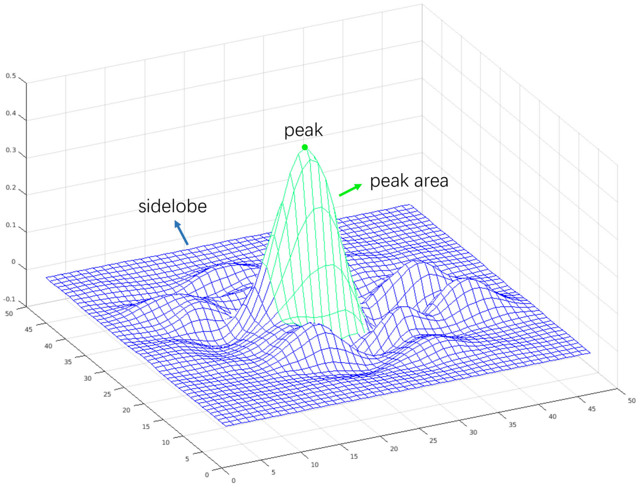


Fig. 2. 3-D visualization of the response map, with peak area and sidelobe marked in green and blue, respectively.

IV. Experiments

A. Experimental Setups

1) Implementation Details: The tracking algorithm is implemented using the Pytorch toolbox on a PC with 3.5-GHz CPUs and a GTX 1080 GPU. For the initialization of *Net-A*, it is trained with the first frame until the loss reaches a threshold of 0.001. The regression target labels are generated using a 2-D Gaussian function with peak value of 1.0, and the variance is set to 0.2 times the width and height of the object. For the initialization of *Net-M*, the iteration number is set to 20. The optical flow calculation is completed using the OpenCV package in python. The gradient descent uses the Adam optimizer, with learning rates of 5e-6 and 5e-5 for *Net-A* and *Net-M*, respectively. During online update, we set th_update to 0.3. The iteration times for *Net-A* and *Net-M* update are set to 5 and 2, respectively.

The width of the square search window is set to four times the maximum value of the target width and height. A grid search experiment on an independent dataset is performed, with the enlargement factor varying from 2 to 5. The results are at a similar level when the factor falls within the range of 2.5–4. Generally, larger search area could provide more negative samples in tracker's initialization training, increasing the discriminative power of the learned model. Moreover, if the target undergoes rapid movement or occlusion, the tracker with a larger search area is more likely to catch up with the target. For the sake of

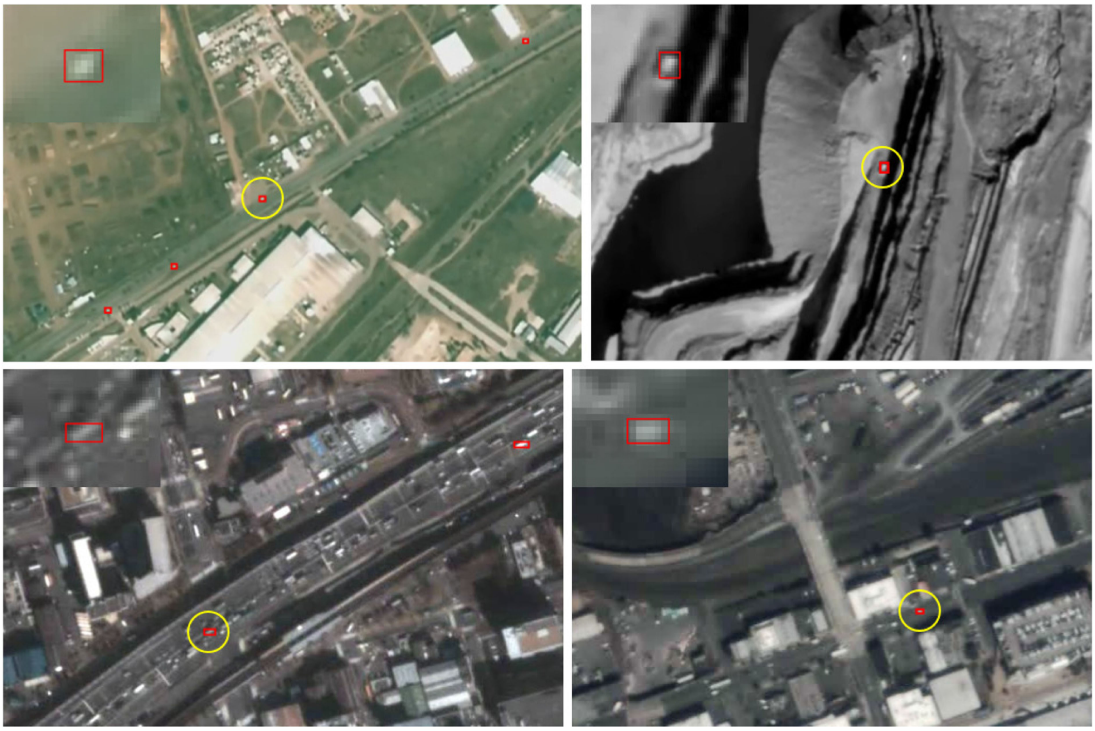


Fig. 3. Example scenes and tracking targets of the satellite videos used in the experiments. In each scene, one selected target is enlarged and displayed in the corners for better view.

more negative training samples during the initialization phase, a relatively larger enlargement factor of 4 is thus adopted.

2) Dataset: The test sequences of the satellite videos are captured by SkySat-1 and Jilin-1 satellites. The satellite videos are cropped into nine small sequences, and 31 moving vehicles are selected as the tracking targets [19]. There are a total of 3161 bounding boxes in this dataset, with video lengths ranging from 70 to 140 frames. Vehicles selected include cars, buses, and even large trucks in mines and ports; thus, the size of the target ranges from around 4 to 21 pixels. For a comprehensive evaluation, more than half of the video clips have challenging attributes such as confused background or similar object nearby. Several targets are shown in Fig. 3 to give an overview of the satellite video dataset used.

3) Compared Methods: Several state-of-the-art trackers are selected for performance comparison, including DSST [8], KCF [7], CF2 [12], ECO [15], STCT [30], DSLT [20], and MDNet [13]. In these methods, KCF and DSST are the classic DCF trackers, while others all use deep learning techniques more or less. Both CF2 and ECO use CNN features, and ECO gives effective improvement to the DCF framework, making it one of the best methods in object tracking. MDNet trains a small-scale network by the multidomain learning strategy. STCT and DSLT are two leading methods based on the CRN. The sizes of the objects in satellite videos are small, but the selected trackers are mostly targeting at general video object tracking tasks. For fair comparison, some settings in their original implementations are modified to fit them for satellite videos.

4) Evaluation Metrics: The VOT2016 [35] toolkit is integrated with the test satellite videos for evaluation. During evaluation, the overlap between the predicted and ground truth bounding boxes in each tracking frame will be recorded. The overlap score S is calculated as follows:

$$S = \frac{|a_g \bigcap a_p|}{|a_g \bigcup a_p|}. \tag{7}$$

where a_g and a_p represent the area of ground truth and predicted bounding box, respectively.

When the overlap becomes zero, one failure is counted, and the tracker will be reinitialized five frames after the failure.

The VOT2016 toolkit provides a reset-based average overlap measure denoted as accuracy (A), which excludes those reinitialization frames from calculation. The per-sequence accuracy A_i is calculated by averaging per-frame overlap score S_t over valid frames

$$A_i = \frac{1}{N_{\text{valid}}} \sum_{t=1}^{N_{\text{valid}}} S_t \tag{8}$$

and the accuracy for the evaluation is the mean of A_i weighted by the length N_i of each sequence (m sequences in total)

$$A = \frac{1}{N_{\text{total}}} \sum_{i=1}^{m} N_i A_i. \tag{9}$$

On the other hand, the robustness (R) measures how many times the tracker loses the target (fails) during tracking. It is represented as the weighted average of per-sequence failure times R_i, as

$$R = \frac{1}{N_{\text{total}}} \sum_{i=1}^{m} N_i R_i. \tag{10}$$

To rank different trackers, VOT provides a measure called expected average overlap (EAO) [36], which combines the raw values of per-frame accuracies and failures. EAO is an estimator of the average overlap a tracker is expected to attain on a large collection of short-term sequences with the same visual properties as the given datasets [35]. In brief, suppose that we could obtain the expected average overlap $\hat{A}_{N_s} = \langle A_{N_s} \rangle$ on a range of sequence lengths, i.e., $N_s = 1 : N_{\max}$; the EAO value is computed as the average of $\hat{A}_{N_s}$ over an interval $[N_{lo}, N_{hi}]$ of typical short-term sequence lengths as

$$\text{EAO} = \frac{1}{N_{hi} - N_{lo}} \sum_{N_s = N_{lo}}^{N_{hi}} \hat{A}_{N_s}. \tag{11}$$

See [36] for more details about the definition and calculation of the EAO measure. Apart from the above three measures, the toolkit also provides a no-reset average overlap, denoted as AO, similar to OTB [1] benchmark, based on tests where the tracker continues to the end without resets. The calculation of AO resembles (8) and (9).

In our experiments, EAO is used as a primary measure, and A, R, and AO are also provided for references. Running speed of the trackers are also provided in terms of frame per second (FPS).

B. Ablation Studies

The CRAM consists of a feature extraction part and a convolutional regression part. Contributions from these components are analyzed through ablation experiments. The performances of different deep features of the VGG16 network are analyzed first. Then, the performance of the CRN is compared with the DCF tracking framework. In the above two studies, only the *Net-A* model is used for fare comparison. In the third group of comparison, the *Net-M* model is inserted to validate the contribution of motion features. And a fixed α of 0.5 is also compared with the adaptive weight strategy used in the CRAM. Besides, an experiment is conducted to show the necessity of upsampling search areas before CNN feature extraction.

1) Feature Selection: Features extracted from different layers of a pretrained CNN encode different visual information. It is generally believed that the deep-layer activations encode semantic information and are robust to significant appearance variation, while shallow layers provide more precise localization but are less invariant to appearance changes. Therefore, the layer used for feature extraction in the CRAM should be carefully selected by experiments. Appearance features extracted from five different convolutional layers of the pretrained VGG16 network are evaluated. From shallow to deep, the selected layers are the second (conv1-2), the fourth (conv2-2), the seventh (conv3-3), the tenth (conv4-3), and the thirteenth (conv5-3) layer. The evaluation results are listed in the first five rows of Table I. First ranks in each column are shown in red. For robustness, a smaller value is better. For other metrics, a larger value is better. As demonstrated in the results, activations of the conv3-3 layer generate the best EAO value. For the features extract from layers shallower or deeper than conv3-3, the corresponding tracking performances decrease to some extent. It can be inferred that this layer keeps enough information for precise localization, as well as good recognition ability.

TABLE I
PERFORMANCE EVALUATIONS OF TRACKERS BUILT ON TOP OF THE CRN AND THE DCF FRAMEWORK, WITH DIFFERENT VISUAL FEATURES

Feature	Base	EAO	A	R	AO
conv1-2	CRN	0.6339	0.69	**0.11**	0.67
conv2-2	CRN	0.6584	**0.70**	**0.11**	**0.68**
conv3-3	CRN	**0.6638**	**0.70**	**0.11**	**0.68**
conv4-3	CRN	0.6395	0.69	**0.11**	0.67
conv5-3	CRN	0.4542	0.69	0.42	0.64
I	DCF	0.3101	0.49	0.70	0.43
HOG	DCF	0.5231	0.67	0.23	0.63
conv3-3	DCF	0.5517	0.67	0.22	0.64

TABLE II
OVERALL PERFORMANCES OF THE CRAM AND ITS MODIFIED VERSION ON THE EXPERIMENTAL SATELLITE VIDEO DATASET

Methods	EAO	A	R	AO
CRAM-O	0.1919	0.47	1.09	0.39
CRAM-A	0.6638	**0.70**	0.11	0.68
CRAM-fix	0.6921	0.69	0.04	0.69
CRAM	**0.7286**	**0.70**	**0.00**	**0.70**

2) Design Validation: In the CRAM, the commonly used DCF framework is replaced by a CRN to solve the linear regression model. To validate the superiority of the CRN, three DCF trackers are constructed, based on the implementation of KCF [7], each using a different feature. Two of them use handcrafted features of grayscale Intensity (I) and HOG, respectively, and the third one uses the deep features of the conv3-3 layer from the VGG16 network.

Experimental results are shown in the lower part of Table I. Comparing the third and eighth rows of Table I, the EAO value has an improvement of 20% when the base framework is changed from the DCF to the CRN. Other evaluation metrics are also improved. This proves that the CRN is more suitable for satellite video object tracking. The superiority of the CRN can be attributed to its ability in detecting objects from complex background, which is obtained by training with enough real object and background samples. Comparing the last three rows of Table I, hand-crafted features will further decrease the performance of the DCF tracker. This proves the superiority of the deep feature.

As claimed in Section III-A1, the size shrinkage of features and corresponding response map caused by CNN feature extraction will bring significant performance degradation. The experimental results about this issue are shown in Table II. The method CRAM-O denotes the variation of the CRAM, in which the features are extracted from original search area image without upsampling. It can be seen that the results of CRAM-O are worse than the CRAM, with the EAO value only 0.1919. In CRAM-O, the spatial size of the response map is one-fourth the size of the original search area. Since the target location is obtained by finding the highest value in the response map, and mapping it to the original image coordinate, the localization

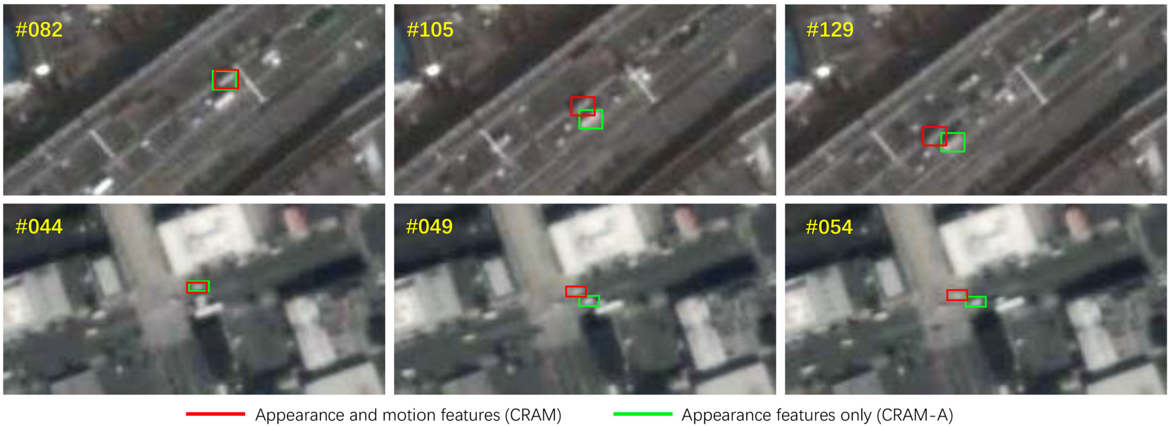

#082
#105
#129
#044
#049
#054
Appearance and motion features (CRAM)
Appearance features only (CRAM-A)

Fig. 4. Performance comparison of CRAM-A, which uses appearance features only, and CRAM, which uses both appearance and motion features.

resolution of CRAM-O is, therefore, 4 pixels of the original image. Considering the small target size in satellite videos, this brings ambiguity about half the size of the ground truth to the predicted target bounding box. Consequently, the tracker will more likely drift from the target. On the contrary, by upsampling search areas before feature extraction to keep the size of response map same as the original image, the tracker could give a more precise target center location.

3) Combination of Appearance and Motion Features: The last ablation study is to analyze the contribution of motion features. Two modified versions of the CRAM are constructed for performance comparison. One is denoted as CRAM-A, in which the *Net-M* part is removed, and only the appearance features are used for tracking. Another is CRAM-fix, in which both *Net-M* and *Net-A* are used, but the value of weighting factor α is fixed to 0.5. It means that the contribution of appearance and motion features are equally important to the final results. The experimental results are shown in Table II. Compared with CRAM-A, the CRAM performs much more robustly, and the failure rate denoted as robustness (R) decreases from 0.11 to 0. In other words, the CRAM could track all the targets in the test video sequences successfully. As a consequence, the EAO value has an increase of about 10%. It proofs the effectiveness of motion features in challenging scenarios. And comparing the performance of CRAM-fix with the CRAM, it can be seen that using an adaptive weighting factor of α can harness the motion information in a superior way and reaches a better performance.

Visualization of tracking results with and without motion features is shown in Fig. 4. It further illustrates how the motion features take effect when appearance features only are not enough. In the two sequences shown in Fig. 4, similar cars are driving near the one on tracking. Using appearance features only, the tracker struggles in the two scenarios and fails to lock the target. However, with rich complementary information provided by motion features, the tracker could identify the target successfully. Fig. 5 illustrates detailed intermediate results of the tracking process for the frame in the middle of the second row in Fig. 4. The first three subfigures are the visualization of the response maps from *Net-A*, *Net-M*, and the final integration. Brighter color represents higher response value. The last subfigure is the image of the enlarged search area. Bounding boxes in red, green, and white, represent tracking results of the CRAM, CRAM-A, and ground truth, respectively. In the response map from *Net-A*, there are two peaks indicating possible target location, and the maximum (denoted by red cross) is on the wrong target. And its PSR is calculated as 8.4. On the contrary, the response map from *Net-M*, which uses motion cues, only has one peak value falling on the true target. Its PSR is calculated as 15.0. In this case, the adaptive weighting factor α imposes *Net-M* more influence to the final response map, such that the final response map can led to a successful prediction. Moreover, in some other cases, as shown in Fig. 6, the CRAM could also benefit from the adaptive weight strategy when predictions from *Net-A* are more reliable. To sum up, the tracker performs robustly with the help of an additional motion features and an adaptive weighting factor α.

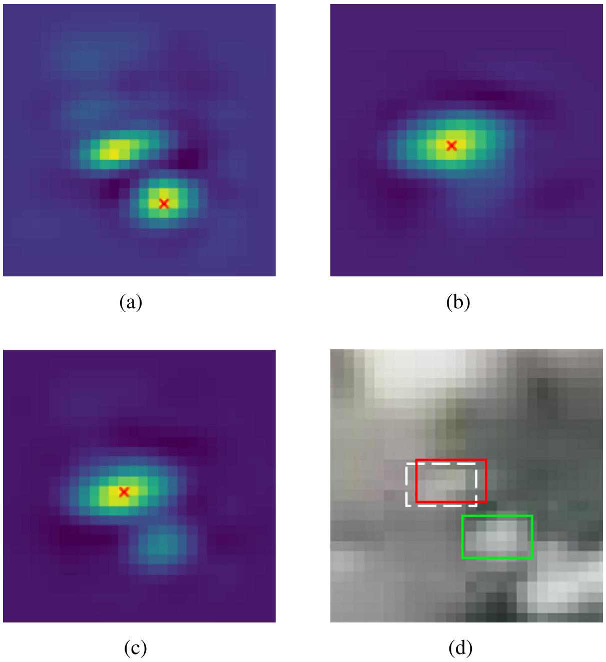


Fig. 5. Visualization of intermediate results during tracking process when *Net-M* is more reliable. (a) Response from *Net-A*, with $\mathrm{PSR}_A = 8.4$. (b) Response from *Net-M*, with $\mathrm{PSR}_M = 15.0$. (c) Integrated response map. (d) Search area image.

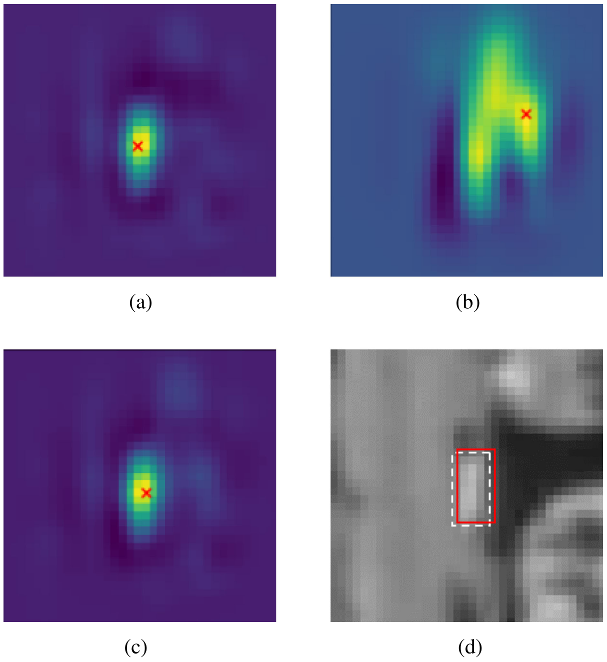


Fig. 6. Visualization of intermediate results during the tracking process when *Net-A* is more reliable. (a) Response from *Net-A*, with $\mathrm{PSR}_A = 35.7$. (b) Response from *Net-M*, with $\mathrm{PSR}_M = 4.9$. (c) Integrated response map. (d) Search area image.

TABLE III
COMPARISON OF THE CRAM WITH OTHER STATE-OF-THE-ART TRACKERS ON THE EXPERIMENTAL SATELLITE VIDEO DATASET

Methods	EAO	A	R	AO	FPS
DSST [8]	0.3761	0.68	0.62	0.56	173.92
KCF [7]	0.5231	0.67	0.23	0.64	**555.49**
DSLT [20]	0.4268	0.61	0.35	0.56	4.02
CF2 [12]	0.5085	0.66	0.22	0.62	1.67
STCT [30]	0.5513	0.66	0.28	0.60	3.15
ECO [15]	0.5870	**0.71**	0.13	0.68	0.81
MDNet [13]	0.6621	0.68	0.06	0.68	1.30
CRAM	**0.7286**	0.70	**0.00**	**0.70**	17.50

C. Comparisons With State-of-the-Art

1) Quantitative Experimental Results: Quantitative experimental results are shown in Table III. The EAO of the CRAM ranks first among all the trackers with its value 0.73, which surpasses the second best tracker MDNet 10%. Besides, the CRAM also exhibits top performances in R and AO metrics. MDNet and ECO trackers reaches close A and AO value as the CRAM, showing their fine generalization ability on various kinds of videos. Nevertheless, their complexities are too high for real-time tracking, and the speed of the CRAM is more than ten times faster. As can be seen from the third column of the table, performance difference in accuracy is not severe among top-ranked trackers. However, the robustness of these methods varies a lot. The good performance of our method owes a great deal to its robustness, which is brought by the combination of appearance and motion features and the regression network framework. The results listed in Table III are also visualized in Figs. 7 and 8. Fig. 7 shows the EAO plot of all the tackers in the experiments in an ascending order, while Fig. 8 shows the A–R (accuracy–robustness) plot of each tracker. In Fig. 8, trackers with better performance should be closer to the upper right corner. The robustness value in this figure is calculated by $\exp(\frac{-FS}{N_{\text{frames}}})$ proposed in [37], where F is failure times and N_{frames} the amounts of frames in total. These results indicate that the CRAM has a better performance comparing with other state-of-the-art trackers on satellite video object tracking.

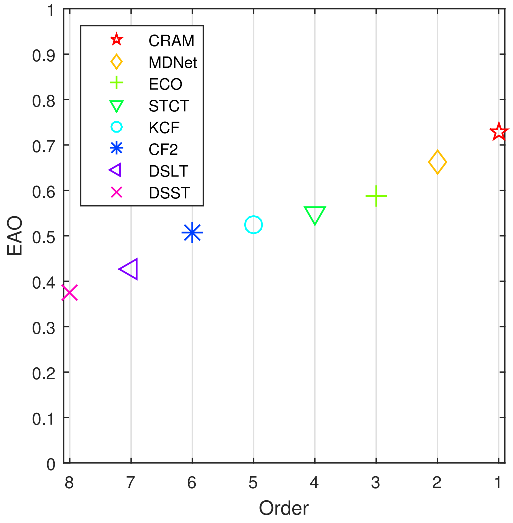


Fig. 7. EAO rank plot of the trackers. The CRAM ranks the first according to the EAO value.

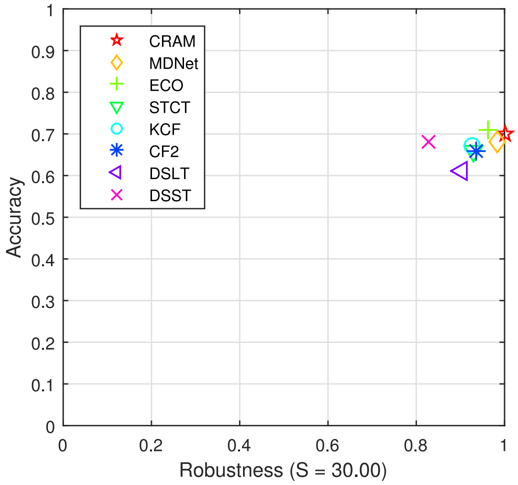


Fig. 8. A–R plot of the trackers. Better trackers should be closer to the upper right corner.

2) Qualitative Experimental Analysis: Fig. 9 shows the experimental results of four top-ranked trackers STCT, ECO, MDNet, and CRAM on three challenging sequences. The CRAM can successfully track the target in all the three sequences, while other trackers encounter tracking failures more or less. In the first sequence, the size of the moving vehicle is very small, only about 4 × 5 pixels. The contrast between the target and the background is not obvious, and a misleading speckle in the road, which has similar shape with the target, stops ECO and STCT trackers from tracking the real vehicle. Fig. 10 shows the intermediate

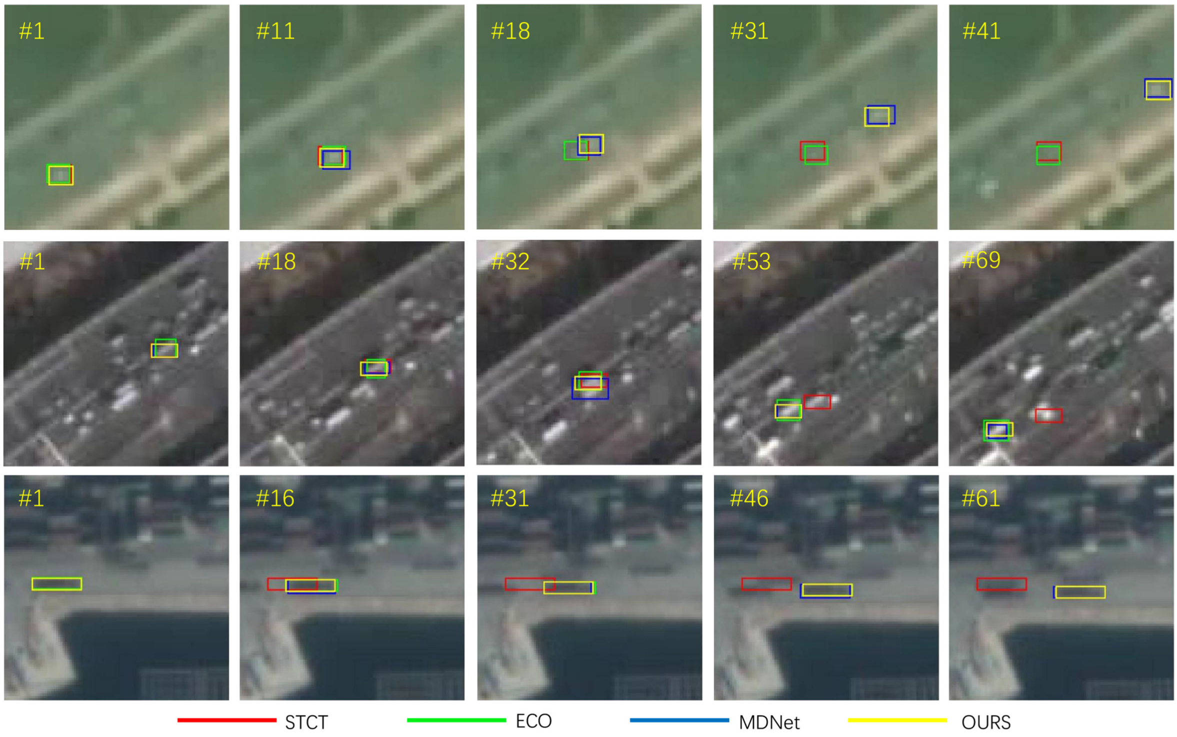
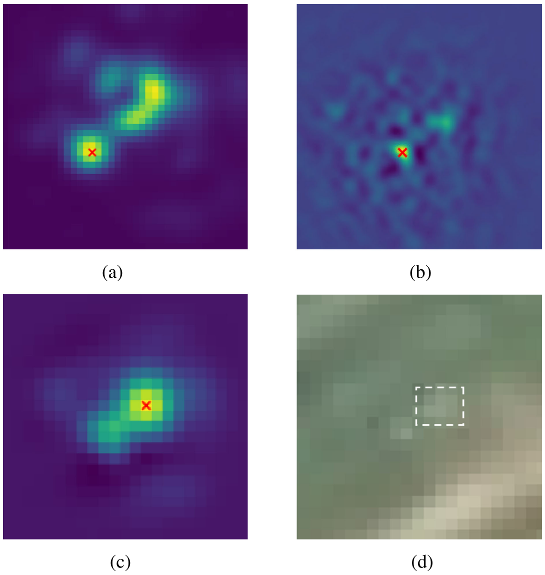


Fig. 9. Comparison of STCT, ECO, MDNet, and CRAM trackers on three challenging satellite sequences.



(a)

(b)

(c)

(d)

Fig. 10. Visualization of intermediate results during the tracking process for the 18th frame in the first video sequence shown in Fig. 9. (a) Predicted heat map of STCT. (b) Continuous score map of ECO. (c) Response map of the CRAM. (d) Cropped original image with ground truth bounding box.

results of STCT, ECO, and CRAM for the 18th frame in the first sequence. Only our CRAM method gives the correct target location prediction. This indicates that the discrimination abilities of STCT and ECO are possibly not strong enough. The CRAM benefits from the advantage of CRN models and, thus, still track the target in the search area. In the second sequence, the target vehicle drives inside the congested traffic. The STCT method is distracted by a similar vehicle passing by. The MDNet method, although not losing the target, is influenced by the vehicle and gives an inappropriate target size estimation. Similar appearance leads to similar feature representations; therefore, these trackers could be misled by the distractors. However, with the help of motion information, the CRAM could lock on the target continuously, as illustrated in Fig. 5. Large trucks with a high aspect ratio are shown in the last sequence. The STCT tracker is not well initialized in the first frame; thus, it cannot track the truck in consequent frames. The reason may be that the training parameters for its model are not suitable in this case. In conclusion, these results prove that the CRAM can provide better tracking performances with complex scenes of satellite videos.

V. Conclusion

In this article, we propose a new method CRAM for object tracking in satellite videos. The CRAM harnesses the power of end-to-end learning through the CRN. Deep convolutional features that bring about effective feature representations are successfully used on satellite videos. With the auxiliary of the motion features, tracking drifts on satellite videos can be alleviated, leading to the improvement of the CRAM in robustness and accuracy. Experiments on the modified benchmark with satellite videos indicate that the CRAM performs favorably against state-of-the-art trackers and also show that the motion information is essential to achieve good tracking results. Future work would be focused on a more in-depth integration method to explore motion cues in solving challenging tracking situations.

REFERENCES

[1] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2411–2418.

[2] G. Kopsiaftis and K. Karantzalos, "Vehicle detection and traffic density monitoring from very high resolution satellite video data," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Milan, Italy, 2015, pp. 1881–1884.

[3] J. Shao, B. Du, C. Wu, and L. Zhang, "Tracking objects from satellite videos: A velocity feature based correlation filter," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 7860–7871, Oct. 2019.

[4] G. Zhang, "Satellite video processing and applications," *J. Appl. Sci.*, vol. 34, no. 4, pp. 361–370, Jul. 2016.

[5] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 2544–2550.

[6] J. F. Henriques, R. Caseiro, P. Martins, and J. P. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proc. 12th Eur. Conf. Comput. Vis.*, 2012, pp. 702–715.

[7] 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.

[8] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. Brit. Mach. Vis. Conf.*, 2014, pp. 65.1–65.11.

[9] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4310–4318.

[10] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, "Staple: Complementary learners for real-time tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1401–1409.

[11] B. Du, Y. Sun, S. Cai, C. Wu, and Q. Du, "Object tracking in satellite videos by fusing the kernel correlation filter and the three-frame-difference algorithm," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 2, pp. 168–172, Feb. 2018.

[12] C. Ma, J. B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 3074–3082.

[13] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4293–4302.

[14] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional siamese networks for object tracking," in *Proc. 14th Eur. Conf. Comput. Vis. Workshops*, 2016, pp. 850–865.

[15] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ECO: Efficient convolution operators for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6931–6939.

[16] J. Wu, G. Zhang, T. Wang, and Y. Jiang, "Satellite video point-target tracking in combination with motion smoothness constraint and grayscale feature," *Acta Geodaetica Et Cartographica Sinica*, vol. 46, no. 9, pp. 1135–1146, Sep. 2017.

[17] J. Shao, B. Du, C. Wu, J. Wu, R. Hu, and X. Li, "VCF: Velocity correlation filter, towards space-borne satellite video tracking," in *Proc. IEEE Int. Conf. Multimedia Expo.*, Jul. 2018, pp. 1–6.

[18] B. Du, S. Cai, and C. Wu, "Object tracking in satellite videos based on a multiframe optical flow tracker," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 8, pp. 2043–3055, Aug. 2019.

[19] Y. Guo, D. Yang, and Z. Chen, "Object tracking on satellite videos: A correlation filter-based tracking method with trajectory correction by Kalman filter," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 9, pp. 3538–3551, Sep. 2019.

[20] X. Lu, C. Ma, B. Ni, X. Yang, I. Reid, and M.-H. Yang, "Deep regression tracking with shrinkage loss," in *Proc. 15th Eur. Conf. Comput. Vis.*, 2018, pp. 369–386.

[21] K. Chen and W. Tao, "Learning linear regression via single-convolutional layer for visual object tracking," *IEEE Trans. Multimedia*, vol. 21, no. 1, pp. 86–97, Jan. 2019.

[22] M. Danelljan, G. Bhat, S. Gladh, F. S. Khan, and M. Felsberg, "Deep motion and appearance cues for visual tracking," *Pattern Recognit. Lett.*, vol. 124, pp. 74–81, Jun. 2019.

[23] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, San Diego, CA, USA, May 2015, pp. 1–14.

[25] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[26] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Dec. 2015, pp. 621–629.

[27] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2018, pp. 8971–8980.

[28] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[29] L. Wang, W. Ouyang, X. Wang, and H. Lu, "Visual tracking with fully convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3119–3127.

[30] L. Wang, W. Ouyang, X. Wang, and H. Lu, "STCT: Sequentially training convolutional networks for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1373–1381.

[31] K. Chen and W. Tao, "Convolutional regression for visual tracking," *IEEE Trans. Image Process.*, vol. 27, no. 7, pp. 3611–3620, Jul. 2018.

[32] Y. Song, C. Ma, L. Gong, J. Zhang, R. W. H. Lau, and M. H. Yang, "CREST: Convolutional residual learning for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2574–2583.

[33] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Proc. Scand. Conf. Image Anal.*, 2003, pp. 363–370.

[34] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *Int. J. Comput. Vis.*, vol. 92, no. 1, pp. 1–31, 2011.

[35] M. Kristan *et al.*, "The visual object tracking VOT2016 challenge results," in *Proc. 14th Eur. Conf. Comput. Vis. Workshops*, Oct. 2016, pp. 777–823.

[36] M. Kristan *et al.*, "The visual object tracking VOT2015 challenge results," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Dec. 2015, pp. 564–584.

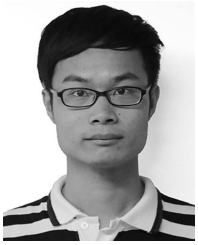
[37] L. Čehovin, M. Kristan, and A. Leonardis, "Is my new tracker really better than yours?" in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Mar. 2014, pp. 540–547.

Zhaopeng Hu received the bachelor's degree in remote sensing science and technology in 2017 from Wuhan University, Wuhan, China, where he is currently working toward the master's degree with the School of Remote Sensing and Information Engineering.

His research interests include object tracking, photogrammetry and remote sensing, and machine learning.

Daiqin Yang (Member, IEEE) received the B.E. and M.E. degrees from the Huazhong University of Science and Technology, Wuhan, China, in 1999 and 2002, respectively, and the Ph.D. degree from the University of Hong Kong, Hong Kong, in 2006, all in electrical and electronic engineering.

She is currently an Associate Professor with the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan. Her research interests include image processing and analysis, photogrammetry and remote sensing, and intelligent systems.



Kao Zhang received the B.Eng. degree in remote sensing science and technology and the M.Eng. degree in surveying engineering from the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, China, in 2014 and 2016, respectively, where he is currently pursuing the Ph.D. degree majored in photogrammetry and remote sensing.

His research interests include computer vision and image/video processing.

Zhenzhong Chen (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the Chinese University of Hong Kong, Hong Kong, in 2007.

He is currently a Professor with Wuhan University, Wuhan, China.