Remote sensing target tracking in satellite videos based on a variable-angle-adaptive Siamese network

Fukun Bi1 | Jiayi Sun1 | Jianhong Han1 | Yanping Wang1 | Mingming Bian2

1 Department of School of Information Science and Technology, North China University of Technology, Beijing, China
2 Qian Xuesen Laboratory, Beijing Institute of Spacecraft System Engineering, Beijing, China

Abstract

Remote sensing target tracking in satellite videos plays a key role in various fields. However, due to the complex backgrounds of satellite video sequences and many rotation changes of highly dynamic targets, typical target tracking methods for natural scenes cannot be used directly for such tasks, and their robustness and accuracy are difficult to guarantee. To address these problems, an algorithm is proposed for remote sensing target tracking in satellite videos based on a variable-angle-adaptive Siamese network (VAASN). Specifically, the method is based on the fully convolutional Siamese network (Siamese-FC). First, for the feature extraction stage, to reduce the impact of complex backgrounds, we present a new multifrequency feature representation method and introduce the octave convolution (OctConv) into the AlexNet architecture to adapt to the new feature representation. Then, for the tracking stage, to adapt to changes in target rotation, a variable-angle-adaptive module that uses a fast text detector with a single deep neural network (TextBoxes++) is introduced to extract angle information from the template frame and detection frames and performs angle consistency update operations on the detection frames. Finally, qualitative and quantitative experiments using satellite datasets show that the proposed method can improve tracking accuracy while achieving high efficiency.

1 INTRODUCTION

The field of remote sensing target tracking in satellite videos [1–4] is gradually developing. Earth observation satellites are a type of detection platform with remote sensing tracking technology. Video payloads are one of the main technologies for satellite detection platforms. For example, the Jilin No. 1 satellite and Skybox satellites are satellites carrying video payloads with the ability to observe highly dynamic targets, and satellite data obtained from them contain abundant dynamic information on the target areas. In particular, the role of remote sensing target tracking in satellite videos is more significant in certain fields, such as military reconnaissance, surveillance of maneuvering targets in large areas, and disaster rescue. However, the characteristics of satellite videos also lead to great difficulty in target tracking, and most state-of-the-art tracking algorithms can effectively solve the tracking problem only for natural scenes. Specifically, remote sensing target tracking in satellite videos is affected by many factors, such as the rotation changes of highly dynamic targets and the presence of many confusing interfering targets due to complex backgrounds, which greatly affect the tracking performance of the existing tracking algorithms. Therefore, effectively designing a robust tracking algorithm is extremely challenging.

At present, target tracking technology has undergone considerable development, and a large number of effective methods have emerged. Among them, correlation-filtering-based tracking algorithms [5–12] and tracking algorithms based on deep learning [13–16] are the most representative. The introduction of correlation-filtering-based tracking algorithms has greatly improved the efficiency of tracking. This type of algorithm achieves target tracking by generating a high response to the target of interest but a low response to background. For example, Martin et al. [5] proposed the DSST algorithm, in which the tracking task is divided into target position prediction and scale estimation and the scale changes of the target are predicted based on the determined target position by learning an independent scale filter. Ma et al. [6] proposed the LCT algorithm,
in which a target redetection mechanism based on training an online random fern classifier is introduced. Martin et al. [7] proposed the SRDCF algorithm, in which a spatial regularization term is introduced to punish the correlation filter coefficients based on the spatial position, thereby reducing the influence of samples near boundaries and generating a low response to background. However, for remote sensing target tracking in satellite videos, most correlation-filtering-based tracking algorithms have difficulty addressing interference from complex backgrounds and similar targets, which affects the tracking accuracy.

The introduction of tracking algorithms based on deep learning has greatly improved tracking accuracy in complex scenes. For example, Davia Held et al. [14] proposed the GOTURN algorithm, in which a convolutional neural network is trained based on input image pairs and outputs the change in the search area relative to the target position in the previous frame. Nam et al. [15] proposed the MDNet algorithm, which uses a pre-trained tracking sequence model to learn a shared representation of the target and performs periodic online updates to adapt to specific domain information. Song et al. [16] proposed the VITAL algorithm, in which a generative adversarial network is introduced to capture changes in target features. However, most of these deep-learning-based algorithms require online fine-tuning mechanisms, which compromise their tracking speed.

In recent years, target tracking algorithms based on Siamese networks [17–22] have gradually attracted the attention of scholars. Such algorithms have the advantage of high tracking accuracy while avoiding the shortcoming of a slow tracking speed. The fully convolutional Siamese network (Siamese-FC) algorithm [17] is one representative algorithm; its core idea is to treat the training of a deep convolutional neural network as a general similarity learning problem in the initial offline phase and then simply evaluate the resulting function online during the tracking process. In this way, Siamese-FC can achieve competitive performance while ensuring strong timeliness. However, due to the complex backgrounds of remote sensing images and the insufficient significance of many features, the desired target can often be easily confused with false alarms nearby. In addition, highly dynamic targets tend to exhibit many rotation changes, but Siamese-FC does not consider the problem of target rotation, making it difficult to maintain long-term stable tracking. To address the above problems, we introduce a new multifrequency feature representation method and a variable-angle-adaptive strategy based on Siamese-FC and propose a corresponding algorithm for remote sensing target tracking in satellite videos based on a variable-angle-adaptive Siamese network (VAASN). The contributions of this article are as follows:

1. For the feature extraction stage, we introduce a new multifrequency feature representation method. To adapt to the new feature representation, we introduce the octave convolution (OctConv) operation into the AlexNet architecture, thereby effectively improving the feature expression capabilities for highly dynamic targets while reducing memory consumption and computational cost.
2. For the tracking stage, we propose a variable-angle-adaptive strategy based on TextBoxes++ to extract the angle information of the target and perform corresponding angle consistency update operations on the detection frames with respect to the template frame, thereby allowing the tracker to adapt to changes in target rotation during tracking.

2 PROPOSED METHODS

In this section, we describe the proposed VAASN algorithm for remote sensing target tracking in satellite videos in detail. As shown in Figure 1, our tracking method is based on Siamese-FC. To improve the feature expression capabilities of the network in the feature extraction stage, we present a new multifrequency feature representation method and introduce the OctConv operation into the AlexNet architecture to adapt to the new feature representation. To improve the adaptability to rotation changes of the target in the tracking stage, we introduce TextBoxes++ to extract the angle information of the target and then perform corresponding angle consistency update operations on the detection frames. Specifically, we rotate the candidate frames in accordance with their angular differences with respect to the template frame and then send them to the detection branch to measure the similarity between the template frame branch and the detection branch. When the target directions in two frames are the same, the response value between them is the highest; thus, the target can be more easily identified, and the impact of frequent changes in target rotation is reduced.

2.1 Siamese-FC

Siamese-FC adopts the traditional multiscale testing method based on enumeration. The network consists of a template branch and a detection branch; the input to the template branch is the tracking target, and the input to the detection branch is a minibatch of candidate frames, which have the same resolution but different scales.

First, the feature map of the template frame, \( \varphi(x_i) \), and the feature maps of the detection frames, \( \{\varphi(x_1), \ldots, \varphi(x_n)\} \), are generated by AlexNet. Then, cross-correlation is used as a similarity measure. The feature map of each detection frame \( \{\varphi(x_1), \ldots, \varphi(x_n)\} \) is individually convolved with the feature map of the template frame \( \varphi(x) \) to calculate the similarity between the two branches at each position in the feature map; thus, a response map set \( \{r_1, \ldots, r_n\} \) is obtained. Finally, the position of the target in the new frame is determined based on the position of the maximum response in the response maps. The formula for calculating the position of the maximum response in the response maps is as follows:

\[
(x, y, m) = \arg \max_{x,y,m} \sum_{m=1}^{n} r_{m} (m = 1 \ldots n)
\]  

where, \((x, y)\) represents the coordinates of the maximum response and \(m\) is the index of the response map in which the maximum response value is found. Then, the corresponding candidate frame is found according to the index value, and the
final tracking result is obtained by combining the coordinates of the maximum response with the scale information of this candidate frame.

In this process, Siamese-FC implements multiscale testing to more accurately determine the proportion of the target and effectively improve the tracking accuracy. However, for remote sensing target tracking in satellite videos, the backgrounds are complex and changeable, making it difficult to identify the target, and the use of the AlexNet architecture for feature extraction in Siamese-FC limits the feature expression capabilities and recognition performance of the network. Moreover, rotation changes of the target can influence the tracking performance, but Siamese-FC does not consider such rotation changes. To solve these problems, we present a multifrequency feature representation for use in the feature extraction stage and introduce a variable-angle-adaptive module for use in the tracking stage to perform angle consistency update operations on the target.

### 2.2 Feature extraction module based on a multifrequency feature representation

Siamese-FC does not update the template online during the tracking process, allowing it to perform fast calculations. However, due to the complex backgrounds of remote sensing images and the presence of many interfering targets, the tracking accuracy may be greatly affected; consequently, the features extracted in Siamese-FC must be sufficiently robust. Therefore, improving the feature representation capabilities of AlexNet is a key issue for Siamese-FC. To address the above problems while greatly reducing memory consumption and computational cost, we introduce a new “plug-in” OctConv operation [23] into the AlexNet architecture [24]. The OctConv operation relies on a multifrequency feature representation in which the output maps of the template and detection frames are each decomposed into a low-spatial-frequency component that retains coarse information and a high-spatial-frequency component that retains detailed information, which are stored in two different channels. Through information sharing between adjacent locations, the length and width of the low-frequency component channel, which contains a large amount of redundant information, are reduced to one-half of those of the high-frequency component channel. In addition, OctConv operations can be directly performed on feature maps containing the two different components. As the resolution of the low-frequency feature maps becomes lower, the receptive field becomes larger. Therefore, when a convolution kernel is used to convolve the low-frequency feature maps, the receptive field is almost doubled, which further helps an OctConv layer to capture long-range contextual information, thereby potentially improving the recognition performance. The specific process of the OctConv operation is shown in Figure 2 below.
We define $X, Y \in \mathbb{R}^{N \times k \times k}$ as the input and output tensors of a convolutional layer and $W \in \mathbb{R}^{N \times k \times k}$ as the matching convolution kernel. The input tensor $X$ can be factored into high- and low-spatial-frequency components as follows: $X = \{X^H, X^L\}$. Similarly, the output tensor $Y$ is naturally factored into $Y = \{Y^H, Y^L\}$, and the convolution kernel $W$ can also be decomposed into $W = \{W^H, W^L\}$ to obtain the high- and low-spatial-frequency components to be convolved with $X^H$ and $X^L$ to construct the output tensor. The output tensor is expressed as:

$$Y^H = Y^{H \rightarrow H} + Y^{L \rightarrow H}$$

$$Y^L = Y^{L \rightarrow L} + Y^{H \rightarrow L}.$$  

(2) 

(3) 

Here, $L \rightarrow H$ and $H \rightarrow L$ represent the information exchange between different frequency components, and $L \rightarrow L$ and $H \rightarrow H$ represent the information update of each frequency component itself. The output tensor can be further expressed as:

$$Y^H = Y^{H \rightarrow H} + Y^{L \rightarrow H} = \text{Conv} \left( \text{avgpool}(X^H, 2), W^{H \rightarrow H} \right)$$

$$+ \text{Conv} \left( X^L, W^{L \rightarrow L} \right)$$

(5) 

In the construction of $Y^H$, to obtain $Y^{L \rightarrow H}$, we first upscale $X^L$ and then perform a traditional convolution operation. For the construction of $Y^{H \rightarrow L}$, if $X^H$ is downsampled with a step size of two, then a corresponding upsampling operation will need to be performed on $Y^{L \rightarrow L}$ before final feature fusion; however, this will cause a certain degree of feature offset, preventing proper alignment of the features during feature fusion, which will affect the performance. Therefore, we instead choose to perform an average pooling operation on $X^H$. By contrast, in the construction of $Y^L$, we uniformly use the traditional convolution method for calculation.

Each feature map in $Y_{p,q} \in \mathbb{R}^N$ can be represented as follows:

$$Y_{p,q} = \sum_{i,j \in N_k} W^H_{p-i,q-j} Y^H(p+i,q+j)$$

$$+ \sum_{i,j \in N_k} W^L_{p-i,q-j} Y^L(p+i,q+j).$$

(7) 

The position coordinates are represented by $(p, q)$, and the neighbourhood coordinates are defined as $N_k = \{(i, j) : i = \{-k-1, \ldots, k\}, j = \{-k-1, \ldots, k\}\}$; thus, the output tensor can be further expressed as:

$$Y_{p,q}^H = Y{p,q}^{H \rightarrow H} + Y{p,q}^{L \rightarrow H} = \sum_{i,j \in N_k} W^H_{p-i,q-j} Y^H(p+i,q+j)$$

$$+ \sum_{i,j \in N_k} W^L_{p-i,q-j} Y^L(p+i,q+j).$$

(8) 

As mentioned above, we introduce the multi-frequency feature representation into the feature extraction stage in Siamese FC, and operate with octave convolution. For octcov convolution, we need to define a scale factor, which represents the proportion of low frequency components. The hyperparameter of the first convolutional layer are set to $a_{in} = 0$, $a_{out} = a$, the hyperparameter of the last convolutional layer are set to $a_{in} = a$, $a_{out} = 0$, and the hyperparameter of the middle hidden layer are set to $a_{in} = a_{out} = a$.

### 2.3 Variable-angle-adaptive module

Siamese-FC adopts the traditional multiscale testing method based on enumeration to solve the problem of changes in the scale of the target. However, Siamese-FC does not consider the case of target rotation and is consequently unable to handle the frequent rotation changes often exhibited by targets in satellite videos. If this problem is ignored, due to the continuity characteristics of the frames, the tracking accuracy will be greatly

----

**FIGURE 2** Structure diagrams of the OctConv operation. (a) Schematic diagram of the OctConv operation in the input layer, the first five kernels perform ordinary convolution operations on the input channels to obtain five normal output channels. The remaining five kernels perform the OctConv operation on the downsampled input channels to obtain another five output channels. (b) Schematic diagram of the OctConv operation in the hidden layers, the first six kernels perform ordinary convolution operations on the input channels, upsample the obtained small feature maps, and finally yield six output channels. The remaining four kernels perform the OctConv operation on the downsampled input channels to obtain another four output channels.
reduced, or the tracking target may even be lost. To address the above problem, we introduce a new angle consistency update operation to compensate for the fact that Siamese-FC does not consider target rotation. Our method performs particularly well on long-term videos with many rotation changes of the target.

Since the template frame in Siamese-FC plays an irreplaceable role in information transfer, we introduce a TextBoxes++ module [25] to extract the angle information of the template frame in preparation for performing angle consistency update operations on the detection frames. After the template frame is sent to the TextBoxes++ module, the textbox layer predicts the target existence probability for each box based on the feature map and outputs a bounding box with the angle information of the target in the template frame.

We use \( h_0 \) to denote the default box that matches the ground truth of the target in the template frame. This default box is represented by its centre point coordinates and its width and height, \( h_0 = (x_0, y_0, w_0, h_0) \), and the threshold box with angle information for regression is represented by the coordinates of the four corner points, \( q_0 = (x_0^1, y_0^1, x_0^2, y_0^2, x_0^3, y_0^3, x_0^4, y_0^4) \), which are determined as follows:

\[
\begin{align*}
x_0^1 &= x_0 - \frac{w_0}{2}, & j_0^1 = y_0 - \frac{h_0}{2} \\
x_0^2 &= x_0 + \frac{w_0}{2}, & j_0^2 = y_0 - \frac{h_0}{2} \\
x_0^3 &= x_0 + \frac{w_0}{2}, & j_0^3 = y_0 + \frac{h_0}{2} \\
x_0^4 &= x_0 - \frac{w_0}{2}, & j_0^4 = y_0 + \frac{h_0}{2}
\end{align*}
\] (9)

The position offset of the output is represented by:\n\( (\Delta x, \Delta y, \Delta w, \Delta h, \Delta \theta_1, \Delta \theta_2, \Delta \theta_3, \Delta \theta_4, \theta) \), and this position offset is used to further obtain the bounding box \( q = (x_1^1, y_1^1, x_1^2, y_1^2, x_1^3, y_1^3, x_1^4, y_1^4) \), which carries the angle information of the target in the template frame. The calculation method is as follows:

\[
\begin{align*}
x_1^1 &= x_0 + \Delta x^1, & n &= 1, 2, 3, 4 \\
y_1^1 &= y_0 + \Delta y^1, & n &= 1, 2, 3, 4
\end{align*}
\] (10)

We use \( \alpha(\alpha \in [0, \pi]) \) to represent the angle between the long edge of the bounding box and the positive direction of the \( x \)-axis, as shown in the schematic diagram of the angle information presented in Figure 3 below, to express the angle information of the template frame. However, in rare cases, the output bounding box will be a square box; therefore, to account for this situation, we define the calculation formula as follows:

\[
\alpha_1 = \begin{cases} 
\frac{\pi}{2}, & x_1 = x_4 \\
\frac{\pi}{2}, & y_1 = y_4 \\
\frac{\pi}{2}, & x_1 = x_4 \\
\frac{\pi}{2}, & y_1 = y_4 \\
\end{cases}
\]

\[
\begin{align*}
\alpha &= \begin{cases}
\alpha_1 + \pi, & \alpha_1 < 0 \\
\alpha_1, & \alpha_1 > 0
\end{cases}
\end{align*}
\] (13)

For the detection frames, we adopt the same strategy as for the template frame. We send the current detection frame to the TextBoxes++ module to obtain the angle information of the target and use \( \beta \) to represent the angle information of the target in this detection frame. Then, the detection frame is rotated in accordance with the angle information of the target in the template frame. We use \( \Delta \theta_n \) to denote the angle by which the detection frame needs to be rotated. At the same time, we consider that the direction change of the target during the tracking process is not easy to determine. Therefore, in most cases, we rotate the detection frame by both \( \Delta \theta_1 = \beta - \alpha \) and \( \Delta \theta_2 = \beta - \alpha + \pi \) to attempt to rotate the target in the detection frame to the same angle as the target in the template frame, and we then send the detection frames rotated by both angles as candidate samples to the detection branch. A diagram of this process is shown in Figure 4 below. Note that in the case that the bounding box of the output is a square box, we rotate the detection frame by a total of four angles, namely, \( \Delta \theta_1, \Delta \theta_2, \Delta \theta_3 = \Delta \theta_1 + \frac{\pi}{2} \) and \( \Delta \theta_4 = \Delta \theta_2 + \frac{\pi}{2} \), and all four rotated results are sent to the detection branch as candidate detection frame samples.

To avoid the possibility of large errors or even failure during the angle information extraction and rotation of the detection frame, which may affect the final tracking result, we also send...
the original detection frame to the detection branch as one of the candidate samples. After the candidate samples have passed through the network, when the target direction in one of the candidate samples is consistent with the target direction in the template frame, the response will be greatly increased. The calculation formula to identify the matching sample is as follows:

\[(x, y, a) = \arg\max_{N} S_{a}\ (a = 1 \ldots N) \quad (14)\]

The final detection frame can thus be selected according to the index value \(a\), and the final tracking result is determined by combining the coordinates of this frame with the corresponding rotation angle and scale information. In this way, the updating of the target for angle consistency is realized throughout the tracking process. Notably, in typical video sequences, the direction of the target does not change abruptly; therefore, we choose to perform the angle consistency update operation only every \(t\) frames. In each frame with an angle consistency update, multiscale detection is not performed; instead, the scale information is selected to be consistent with the previous frame. When there is no target frequently rotating, the angle consistency operation frequency can be reduced, that is, the value of \(t\) can be increased. By contrast, the remaining frames are subjected only to multiscale testing, and the rotation angle information is not updated. At this stage, \(N\) represents the number of scale ratios. When there is no frequent change of the target scale, the value of \(N\) can be reduced. This strategy compensates for the fact that Siamese-FC does not consider target rotation to allow better adaptation to frequent rotation changes of the target during the tracking process.

3 | EXPERIMENTS AND DISCUSSION

In this section, to comprehensively evaluate the tracking performance of our method, we qualitatively and quantitatively compare our method with other state-of-the-art algorithms. We also report ablation experiments conducted to verify the effects of the enhanced feature extraction module and the variable-angle-adaptive module on the overall performance.

3.1 Datasets and evaluation metrics

We comprehensively evaluated the tracking performance of the algorithms by using two satellite remote sensing datasets. Dataset 1 consists of 80 long-term video sequences with an average of 28 s per video: 40 Jilin-1 satellite video sequences, 20 Skybox satellite video sequences and 20 public video sequences from the Remote Sensing Image Sparse Representation and Intelligent Analysis competition. This dataset contains a wide variety of targets including planes, ships and contains scenes with challenging characteristics such as complex and changeable backgrounds and frequent changes in target rotation. In addition, we selected 30 of the video sequences with an average of 37 s per video from dataset 1 to form dataset 2, all of which contain the same targets in dataset 1, and contain the challenging scenes with frequent changes in target rotation.

3.2 Ablation experiment

We used long-term satellite remote sensing dataset 1 to conduct ablation experiments to verify the importance of the enhanced feature extraction module and the variable-angle-adaptive module in improving the overall performance. Ablation experiment results are shown in Table 1. It is worth noting that we adopt the same training method as Siamese fc, use the MatConvNet architecture, use the logistic loss function when training the model, and minimize the loss function through batch SGD to obtain the best model. In these experiments, we use a tenfold cross-validation method, which divides the video sequences of dataset 1 into ten parts, and then selects nine (72 video sequences) of them in turn as the training set, and the remaining one (eight video sequences) as the test set, the test is repeated ten times, and the result is the average of the ten test results.

According to the above results, when only the enhanced feature extraction module is added to Siamese-FC, the precision rate is increased from 58.2 to 63.4, the success rate is increased from 47.1 to 52.6, and the speed is increased from 68 to 77 frames per second (FPS). The improvements in the precision rate and success rate can be attributed to the new multifrequency feature representation and the matching OctConv con-
olution, which effectively improve the feature expression capabilities for highly dynamic targets and promote the improvement of the overall recognition performance of the network. In addition, the reduced resolution of the low-frequency maps effectively reduces the demands in terms of storage and computation.

When only the variable-angle-adaptive module is added to Siamese-FC, the precision rate is increased from 58.2 to 64.6, the success rate is increased from 47.1 to 53.4, and the speed is slightly reduced, from 68 to 56 FPS. These improvements in the precision rate and success rate can be attributed to the angle consistency update operation, which effectively compensates for the effect of changes in target rotation. The slight decrease in speed is due to the need to periodically extract the angle information of the target in the detection frames and perform corresponding rotations, which increases the overall amount of calculation performed by the network. In some application scenarios, if there is no frequent rotation of the target under high-speed motion, we can reduce the frequency of angle consistency operations. That is, increasing the value of t can greatly improve efficiency.

Finally, we can see that when both modules are added, the precision rate is increased from 58.2 to 68.1 and the success rate is increased from 47.1 to 56.8, both representing significant improvements compared to Siamese-FC, while the speed is slightly increased from 68 to 73 FPS. These improvements are due to the collaborative influence of the enhanced feature extraction module and the variable-angle-adaptive module.

From these ablation experiments, the impacts of the two proposed modules on the overall performance can be intuitively seen. The results show that the addition of both modules greatly improves the precision rate and success rate of Siamese-FC while also slightly increasing the tracking speed of the network. Especially on long-term satellite remote sensing datasets, the resulting algorithm shows improved performance. Although the addition of the variable-angle-adaptive module somewhat hinders the speed of the algorithm, Siamese-FC already offers good time performance, and the addition of the enhanced feature extraction module enables a considerable speed improvement. On the whole, the efficiency of our method can meet the needs of practical applications, and in the next section, we will visually compare our method with other state-of-the-art algorithms in terms of speed. In addition, our method offers significantly improved performance under changes in target rotation; such rotation changes are commonly encountered in target tracking in satellite videos, and compensating for them is the key to achieving long-term stable tracking.

3.3 Performance comparison with other state-of-the-art tracking algorithms

We evaluated the tracking algorithms on two datasets: dataset 1, a long-term satellite remote sensing dataset consisting of 80 fully annotated videos, and dataset 2, consisting of 30 fully annotated videos selected from dataset 1. It is worth noting that we adopt the same training method as Siamese fc, use the MatConvNet architecture, use the logistic loss function when training the model, and minimize the loss function through batch SGD to obtain the best model. In these experiments, we take a tenfold cross-validation method, that is, the video sequences of the dataset 1 is divided into ten parts, and one of them (eight video sequences) is taken as the testset in turn. A total of ten tests are performed, and the results are taken average value. In these experiments, we take a tenfold cross-validation method, that is, the video sequences of the dataset 1 is divided into ten parts, and one of them (eight video sequences) is taken as the testset in turn. A total of ten tests are performed, and the results are taken average value.

First, we selected representative scenes with complex backgrounds and rotation changes of the targets. We used these representative scenes to visualize the results of ECO-HC, SRDCF, KCF, STAPLE, MDNet, CFNet, Siamese-FC and our method to enable an intuitive qualitative evaluation of the tracking performance of each tracking algorithm. The visualization results are shown in Figures 5–8.

**FIGURE 5** Original image of target rotation changes against a simple background
Our tracking targets are typical moving targets in remote sensing videos, such as airplanes, ships, and cars. The problems we solve are mainly based on the first two categories; this article focuses on airplane and ship as examples of typical visual tracking results. In the above figures, it can be seen that sequence (a) involves target rotation changes against a simple background. When the direction of the target changes slightly, each tracking algorithm can still accurately locate the target. However, when the direction of the target changes significantly, the KCF algorithm loses the target, and the STAPLE, SRDCF, ECO-HC, Siamese-FC, CFNet and MDNet algorithms all produce different degrees of deviation in the tracking results, which affect the accuracy of tracking target positioning. Sequence (b) involves target rotation changes against a complex background; the features of the target are not obvious, and the target and the surrounding gray background can be easily confused. Consequently, the KCF algorithm and the STAPLE algorithm lose the target. Regarding the robustness and adaptability to the continuous rotation of the target demonstrated by the other algorithms considered for comparison, including SRDCF, ECO-HC,
Siamese-FC, CFNet and MDNet, they all show varying degrees of drift, and some eventually lose the target. Thus, it is intuitively found that our method shows superior performance on datasets that contain scenes with challenging characteristics such as complex backgrounds and target rotation changes.

Next, we consider the precision rate and success rate to quantitatively evaluate our method, KCF, STAPLE, SRDCF, ECO-HC, Siamese-FC, CFNet and MDNet. Figures 9 and 10 present the results of the eight selected algorithms on the two dataset.

Finally, we record the precision rates, success rates, and speeds of each of the eight algorithms on the two different datasets in the Table 2 and Table 3 to enable a horizontal comparison of the tracking performance of our method.

As shown above, the precision rate and success rate of our method are higher than those of the other algorithms, and our method also has an advantage in speed. It performs well on long-term satellite remote sensing datasets that contain scenes with challenging characteristics such as complex backgrounds and changes in target rotation. As seen from our comprehensive analysis, this good performance is due to the introduction of our enhanced feature extraction module and variable-angle-adaptive module on the basis of Siamese-FC. Specifically, the enhanced feature extraction module effectively improves the feature expression capabilities for highly dynamic targets and the recognition performance of the overall network; it solves
the problems caused by complex and changeable backgrounds in satellite videos, such as insufficient significance of the target features and the resulting ease with which the target can be confused with similar interfering objects, which seriously affect the tracking accuracy. At the same time, we can intuitively see from Figures 3 and 4 and Table 3 that the variable-angle-adaptive module effectively enhances the robustness and adaptability of our method. On dataset 2, which contains frequent rotation changes, the precision rate and success rate of our method are higher than those of the other state-of-the-art tracking algorithms, further illustrating the effectiveness of the target angle consistency update operation performed in this module, which is reflected in the reduced drift and target loss caused by frequent changes in target rotation during the tracking process.

The variable-angle-adaptive module also causes a certain loss in tracking speed; however, the tracking speed of Siamese-FC is already satisfactory, and the OctConv operation in the enhanced feature extraction module markedly reduces the internal time consumption of the algorithm. Consequently, our method still guarantees a high tracking speed.

### 4 CONCLUSIONS

In this paper, we propose an effective tracking algorithm called remote sensing target tracking in satellite videos based on a variable-angle-adaptive Siamese network (VAASN). Our tracking targets are typical moving targets in remote sensing videos, such as airplanes, ships, and cars. The problems we solve are mainly based on the first two categories. Through qualitative and quantitative performance comparisons, we show that the precision rate and success rate of our method are better than those of other state-of-the-art tracking algorithms, while the tracking speed is also maintained at a high level. The results intuitively verify the effectiveness of our method, especially on long-term datasets that contain scenes with challenging characteristics such as complex backgrounds and target rotation changes. However, our method will fail when the target size is too small or the target features are very unobvious. Therefore, stable tracking of smaller targets will be the focus of our future work.

### REFERENCES

1. Du, B., et al.: Object tracking in satellite videos by fusing the kernel correlation filter and the three-frame-difference algorithm. IEEE Geosci. Remote Sens. Lett. 15(2), 168–172 (2018)
2. Shao, J., et al.: Tracking objects from satellite videos: A velocity feature based correlation filter. IEEE Trans. Geosci. Remote Sens. 57(10), 7860–7871 (2019)
3. Xuan, S., et al.: Object tracking in satellite videos by improved correlation filters with motion estimations. IEEE Trans. Geosci. Remote Sens. 58(2), 1074–1086 (2020)
4. Guo, Y., Yang, D., Chen, Z.: Object tracking on satellite videos: A correlation filter-based tracking method with trajectory correction by Kalman filter. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 12(9), 3538–3551 (2019)
5. Danelljan, M., et al.: Accurate scale estimation for robust visual tracking. In: British Machine Vision Conference, Nottingham, UK (2014)
6. Ma, C., et al.: Long-term correlation tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, Massachusetts (2015)
7. Danelljan, M., et al.: Learning spatially regularized correlation filters for visual tracking. In: 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, pp. 4310–4318 (2015)
8. Xue, X., et al.: Robust correlation tracking for UAV videos via feature fusion and saliency proposals. Remote Sens. 10, 1644 (2018)
9. Henriques, J.F., et al.: High-speed tracking with kernelized correlation filters. IEEE Trans. Pattern Anal. Mach. Intell. 37(3), 583–596 (2015)
10. Lukezie, A., et al.: Discriminative correlation filter with channel and spatial reliability. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, pp. 4847–4856 (2017)
11. Danelljan, M., et al.: ECO: Efficient convolution operators for tracking. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, pp. 6931–6939 (2017)
12. Bertinetto, L., et al.: Staple: Complementary learners for real-time tracking. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, pp. 1401–1409 (2016)
13. Zhang, W., et al.: Coarse-to-fine UAV target tracking with deep reinforcement learning. IEEE Trans. Autom. Sci. Eng. 16(4), 1522–1530 (2019)
14. Held, D., Thrun, S., Savarese, S.: Learning to track at 100 FPS with deep regression networks. In: 2016. European Conference on Computer Vision (ECCV), Amsterdam, (2016)
15. Nam, H., Han, B: Learning multi-domain convolutional neural networks for visual tracking. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, pp. 4293–4302 (2016)
16. Song, Y., et al.: VITAL: Visual tracking via adversarial learning. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, Utah, pp. 8990–8999 (2018)
17. Bertinetto, L., et al.: Fully-convolutional Siamese networks for object tracking. In: The 14th European Conference on Computer Vision, Amsterdam (2016)
18. Valmadre, J., et al.: End-to-end representation learning for correlation filter based tracking. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, pp. 5000–5008 (2017)
19. He, A., et al.: Towards a better match in Siamese network based visual object tracker. In: European Conference on Computer Vision Springer, Cham, (2018)
20. Li, B., et al.: High performance visual tracking with Siamese region proposal network. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, Utah, pp. 8971–8980 (2018)

### ACKNOWLEDGEMENTS

This work was supported in part by the National Natural Science Foundation of China (Grant No. 61971006), Natural Science Foundation of Beijing Municipal (No. 4192021).

### ORCID

Jiayi Sun  https://orcid.org/0000-0002-2212-1351
21. Lee, D.: One-shot scale and angle estimation for fast visual object tracking. IEEE Access 7, 55477–55484 (2019)
22. Qi, Y., et al.: Hedging deep features for visual tracking. IEEE Trans. Pattern Anal. Mach. Intell. 41(5), 1116–1130 (2019)
23. Chen, Y.P., et al.: Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution. In: Proceedings of the 2019 IEEE International Conference on Computer Vision (ICCV), Seoul, Korea (2019)
24. Krizhevsky, A., Sutskever, I., Hinton, G.: ImageNet classification with deep convolutional neural networks. In: Proceedings of the International Conference on Neural Information Processing Systems, Lake Tahoe, Nevada pp. 1097–1105 (2012)

How to cite this article: Bi F, Sun J, Han J, Wang Y, Bian M. Remote sensing target tracking in satellite videos based on a variable-angle-adaptive Siamese network. IET Image Process. 2021;15:1987–1997. https://doi.org/10.1049/ipr2.12170