remote sensing

MDPI

Article

Object Tracking in Satellite Videos Based on Correlation Filter with Multi-Feature Fusion and Motion Trajectory Compensation

Yaosheng Liu [1,2], Yurong Liao [2], Cunbao Lin [2,*], Yutong Jia [3], Zhaoming Li [2] and Xinyan Yang [2]

[1] Graduate School, Space Engineering University, Beijing 101416, China; lys196245@163.com
[2] Department of Electronic and Optical Engineering, Space Engineering University, Beijing 101416, China; 18037470304@163.com (Y.L.); lizhaomingzbxy@163.com (Z.L.); yangxyyz@163.com (X.Y.)
[3] Department of Surveying and Mapping and Space Environment, Space Engineering University, Beijing 101407, China; jiayutong@st.btbu.edu.cn
* Correspondence: cunbaolin@163.com

Abstract: As a new type of earth observation satellite approach, video satellites can continuously monitor an area of the Earth and acquire dynamic and abundant information by utilizing video imaging. Hence, video satellites can afford to track various objects of interest on the Earth's surface. Inspired by the capabilities of video satellites, this paper presents a novel method to track fast-moving objects in satellite videos based on the kernelized correlation filter (KCF) embedded with multi-feature fusion and motion trajectory compensation. The contributions of the suggested algorithm are multifold. First, a multi-feature fusion strategy is proposed to describe an object comprehensively, which is challenging for the single-feature approach. Second, a subpixel positioning method is developed to calculate the object's position and overcome the poor tracking accuracy difficulties caused by inaccurate object localization. Third, introducing an adaptive Kalman filter (AKF) enables compensation and correction of the KCF tracker results and reduces the object's bounding box drift, solving the moving object occlusion problem. Based on the correlation filtering tracking framework, combined with the above improvement strategies, our algorithm improves the tracking accuracy by at least 17% on average and the success rate by at least 18% on average compared to the KCF algorithm. Hence, our method effectively solves poor object tracking accuracy caused by complex backgrounds and object occlusion. The experimental results utilize satellite videos from the Jilin-1 satellite constellation and highlight the proposed algorithm's appealing tracking results against current state-of-the-art trackers regarding success rate, precision, and robustness metrics.

Keywords: satellite videos; object tracking; correlation filter; multi-feature fusion; subpixel positioning; adaptive Kalman filter

check for updates

Citation: Liu, Y.; Liao, Y.; Lin, C.; Jia, Y.; Li, Z.; Yang, X. Object Tracking in Satellite Videos Based on Correlation Filter with Multi-Feature Fusion and Motion Trajectory Compensation. *Remote Sens.* **2022**, *14*, 777. https://doi.org/10.3390/rs14030777

Academic Editors: Michalis Savelonas and Emmanuel Vassilakis

Received: 29 December 2021
Accepted: 4 February 2022
Published: 7 February 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Object tracking is one of the essential methods for dynamic object observation in computer vision, which has been widely used in video surveillance, automatic navigation, artificial intelligence, and other applications [1]. The purpose of object tracking is to predict the object's size and position in subsequent frames based on the initial frame of a video sequence [2]. With the continuous development of commercial remote sensing satellites, such as the Jilin-1 and Zhuhai-1 satellite constellations, high-resolution videos through video satellites are an affordable method of gazing and observing a specific Earth's area to obtain rich information. The Jilin-1 video satellite was launched by China Chang Guang Satellite Technology Co., Ltd and provided 4k high-resolution imagery, capturing detailed information about an area. Indeed, the satellite imagery was about 1-m resolution at 30 frames per second, and therefore object tracking in such satellite videos has gradually become a new research direction. The corresponding practical applications involve traffic vehicle tracking [3], monitoring seawater [4], monitoring natural disasters [5], military reconnaissance, and precision guidance.

Object tracking in a satellite video is widely applied in many fields and explicitly analyzing the object's motion laws according to its trajectory is of great significance. Most moving objects in satellite videos are ships, vehicles, and airplanes, as these are common objects in satellite videos with significant research value. Shao et al. [5] proposed a velocity correlation filter (VCF) algorithm by employing the velocity features with an inertial mechanism (IM) and constructing a specific kernelized correlation filter for object tracking in satellite videos. This method has an appealing performance in single background, but the method's effectiveness in complex backgrounds or similar objects is yet unknown. Moreover, in 2019 the authors further suggested a hybrid kernel correlation filter (HKCF) tracker that adaptively used two complementary features in the ridge regression framework, combined with an adaptive fusion strategy exploiting both features in various videos [6]. Nevertheless, this method is also only suitable for simple background cases, and the optical flow is quite sensitive to illumination variations. In general, the HOG features play a significant role without emphasizing the response of the optical flow features. Du et al. [7] constructed a robust tracker combining the KCF tracker with a three-frame difference algorithm to overcome the difficulty of similar object interference and fewer object features. This method only considers a single background and does not consider the existence of complex background and object occlusions. Moreover, tracking drift can occur when the object is occluded, resulting in object tracking failure. Guo et al. [8] developed a tracker based on a high-speed correlation filter (CF) for object tracking in satellite videos. This technique utilized the global motion features of the moving object in the satellite videos to constrain the tracking process, which is achieved by applying a Kalman filter (KF) to correct the moving object's tracking trajectory. Although this method has dramatically improved performance and tracking accuracy, the Kalman filter must provide convergence before combining it with the KCF algorithm. Therefore, object tracking in the first few frames still relies on the KCF algorithm, and thus the tracking accuracy does not improve much. Xuan et al. [9] solved the object occlusion problem and reduced boundary effect during motion estimation by combining a correlation filter with a Kalman filter. However, in the case of similar objects or complex object backgrounds, the tracking effectiveness of this method needs to be experimentally verified. However, objects in satellite videos are different from traditional objects, and tracking them is challenging, e.g., the object of interest comprises only a few pixels, and therefore similar objects raise the tracking difficulty. Moreover, objects occluded by clouds or other buildings impose tracking drifts. Hence, primarily, the algorithm must overcome object occlusion (Figure 1).

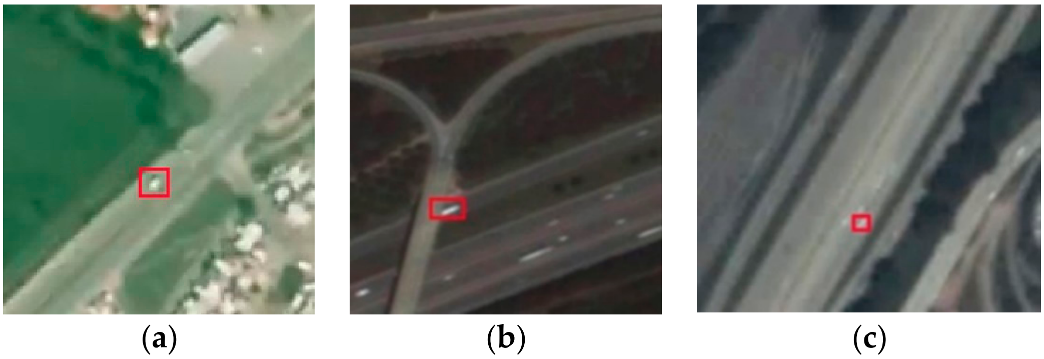


Figure 1. Satellite videos. (**a**) small object size is about 10 × 12 pixels. (**b**) the moving object is occluded. (**c**) the object is similar to its background.

Recently, researchers have introduced the idea of correlation filtering into the object tracking algorithm. The earliest minimum output sum of squared error (MOSSE) [10] algorithm utilized the most basic correlation filtering idea. The circulant structure kernel (CSK) [11] correlation filter algorithm modifies MOSSE, while the kernelized correlation filter (KCF) [12] algorithm combines the advantages of MOSSE and CSK to improve the performance further. This kind of algorithm [13–17] reduces time complexity through a fast

Fourier transform and utilizes cyclic shift samples for sufficient training to enhance object tracking speed and accuracy. The C-COT [18] algorithm utilizes the deep neural network VGG-Net to extract features, interpolates feature maps of different resolutions into the continuous space domain utilizing cubic interpolation, and then finds the object position with subpixel accuracy employing the Hessian matrix. The ECO [19] algorithm improves C-COT and attains an appealing object tracking by reducing the model parameters through feature-subset dimensionality reduction, merging similar sample sets, and sparse updating strategies. The innovative improvement of the ECO algorithm substantially enhances the tracking effect. The correlation filter algorithms belong to discriminative object tracking methods, which have recently become a mainstream research direction due to their high speed and accuracy [20–25]. Since an object's deformation in satellite videos is not apparent, it can be considered that there is almost no deformation, and thus we utilize the KCF algorithm to track the moving object in the satellite videos. However, simply applying the KCF algorithm presents the following shortcomings. First, when the object is occluded, tracking drifts, leading to tracking failure. Second, utilizing dense sampling and Fourier transform causes the tracker to produce boundary effects, affecting tracking accuracy. Third, similar objects may be regarded as real objects, affecting object tracking.

This paper proposes multi-feature fusion and motion trajectory compensation methods to solve poor tracking accuracy. Specifically, we build a robust tracker that fuses the object’s various response features, improves object location accuracy through subpixel positioning, and relies on the adaptive Kalman filter to correct the tracking results of the correlation filter.

More precisely, the main contributions of this paper are as follows:

(1) We propose a multi-feature fusion method to enhance the object features’ expressiveness and improve object tracking in satellite videos. The features employed are the histogram of oriented gradient (HOG) and the convolutional neural networks (CNN) features.

(2) A quadratic parabolic model is proposed to fit the discrete object response values. This scheme approximates the discrete response map to a continuous response map, and based on this, a Taylor series method is used to obtain subpixel position accuracy. This strategy solves the subpixel localization accuracy problem of moving objects.

(3) We reduce the estimation error caused by the noise covariance in the randomly selected Kalman filter (KF) by proposing an adaptive adjustment of both covariances using the state discriminant method (SDM) that affords an increased convergence speed to correct the tracking results of the correlation filter quickly.

2. Materials and Methods

This section briefly reviews the basic theory of the KCF algorithm and then elaborates how to improve the object tracking accuracy by adaptively fusing the features' responses and combining two filters. The KCF algorithm [12] is widely used in object tracking, and its ideas and methods have been presented in many papers, so it will not be elaborated in detail. The proposed method's architecture is illustrated in Figure 2. Initially, we extract the HOG and VGG features from the current frame T, select robust features from the VGG ones, fuse them in parallel with a specific proportion, and obtain the response through the trained correlation filter. Second, we use the designed fusion strategy to fuse the response patch of both filters. Third, we conduct occlusion detection and use the predicted result of the adaptive Kalman filter as the final tracking result if the object is occluded or obtain the object’s subpixel location if it is not occluded. Finally, the noise covariance of the Kalman filter is adjusted according to the Euclidean distance between the center position of the object in the adjacent frames so that the predicted results can compensate and correct for the tracking results of the correlation filter, thus obtaining an accurate object position.

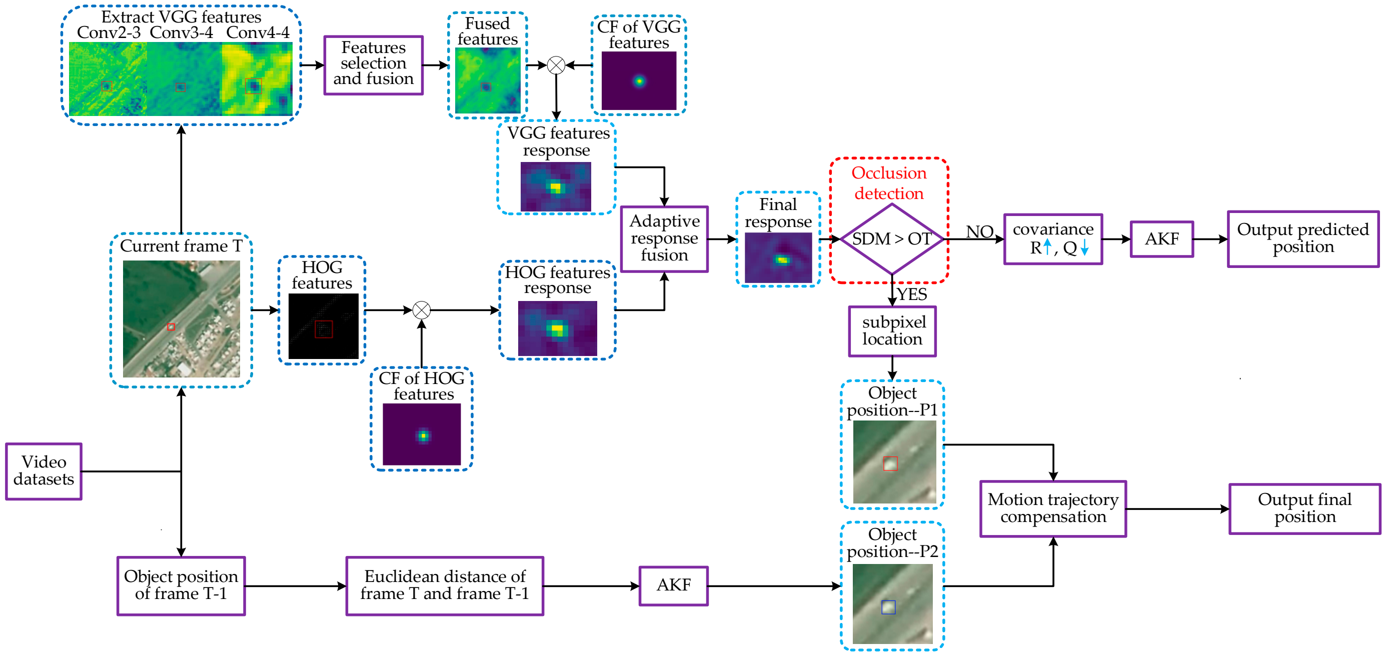


Figure 2. Flowchart of the proposed algorithm. CF and AKF are the abbreviations of correlation filter and adaptive Kalman filter. The Euclidean distance is the object center position of adjacent frames.

2.1. Multi-Feature Fusion

Selecting the object features is crucial, as it directly affects the object tracking results. This paper extracts the object's hand-crafted and deep features for adaptive feature fusion to enhance the object's description and improve the tracking algorithm performance in a complex background.

Considering the hand-crafted features, this paper employs HOG features, mainly containing texture information with high spatial resolution and high target localization accuracy. However, solely relying on HOG features does not afford accurate target tracking in complex background scenarios. Considering depth features for complex backgrounds and illumination variation cases, CNN features have rich texture information and stronger robustness but present low spatial resolution and localization accuracy for the targets. Therefore, this paper fuses the hand-crafted and deep feature response to fully utilize their complementary nature and improve the filter's classification ability. Hence, we rely on the VGG network to extract the object's in-depth features. VGG is a model proposed by Oxford University in 2014 that has demonstrated appealing results in image classification and target detection tasks. Specifically, each image is input into VGG, and its convolutional layers provide the deep features.

Deep convolutional neural networks can extract rich features in spatial and semantic information. The low-level convolutional features contain rich spatial information, and the high-level convolutional features contain rich semantic information. The layered convolutional correlation filter tracking algorithm [26] uses the features extracted from the conv3-5, conv4-5, and conv5-5 layers of VGG-19 to predict the target location. Precisely, the features in the correlation filtering framework weight the response maps of the three layers, achieving a significant improvement in tracking accuracy compared with the traditional feature correlation filtering algorithm.

Therefore, the VGG-19 network is experimentally validated by selecting low, middle, and high-level convolutional layers that contain spatial and semantic information, i.e., conv2-3, conv3-4, and conv4-4, depicted in Figure 3. Although the features extracted from the fifth convolutional layer are semantically rich, these do not further improve the object's description (see Figure 3e), and thus we extract features from the convolutional layer. Nevertheless, the large dimensionality of the deep convolutional layer features increases the computational complexity, especially when using multiple convolutional layer features during training the filters, resulting in slower tracking. Additionally, the compelling convolutional features differ depending on the tracking scenarios. Figure 4 illustrates the CNN layer-2 features of Figure 3a, where some features contribute less or are even invalid for the tracking task, reducing the tracking performance and slowing down the tracking speed. Hence, the appropriate convolutional channels need to be selected considering speed and robustness. Thus, the features are filtered by the ratio of the object and the object search region variances:

$$V_1 = \frac{1}{h_1 \times w_1} \sum_{i,j} \left(E_{i,j} - E \right)^2 \tag{1}$$

$$V_2 = \frac{1}{h_2 \times w_2} \sum_{m,n} \left(T_{m,n} - T \right)^2 \tag{2}$$

so the variance ratio (VR) can be expressed as:

$$VR = \frac{V_1}{V_2} \tag{3}$$

where $E_{i,j}, T_{m,n}$ are the pixel values per layer of the object region and the object search region, respectively; and $\overline{E}, \overline{T}$ are the average pixel values of the object region and the object search region. i, j and m, n are the horizontal and vertical coordinates of the object region and the object search region, respectively. h_1, w_1 and h_2, w_2 are the height and width of the object and the object search region.

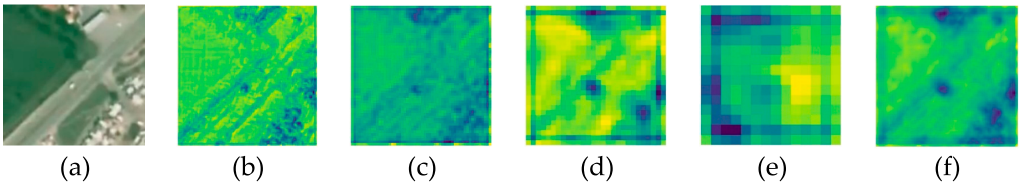


Figure 3. Feature visualization of various layers from the VGG-19 Networks. (**a**) Input image. (**b**) Conv2-3 layer features. (**c**) Conv3-4 layer features. (**d**) Conv4-4 layer features. (**e**) Conv5-4 layer features. (**f**) fused image features.

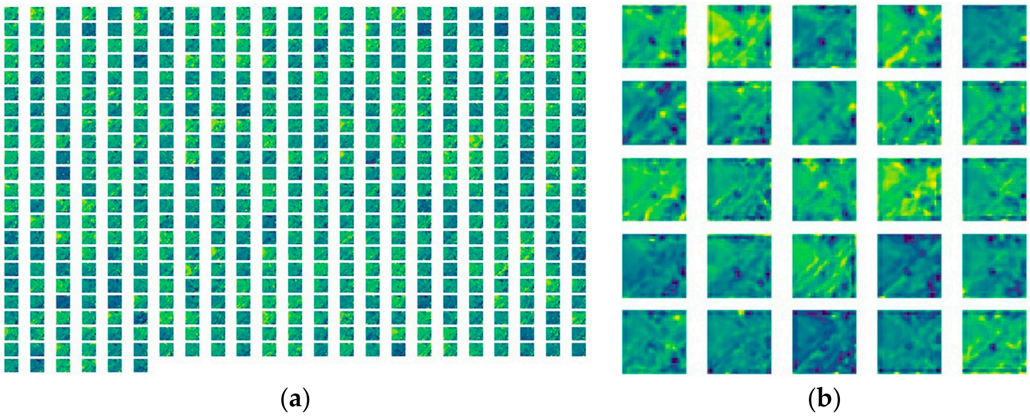


Figure 4. Feature visualization of small objects. (**a**) 256 channels' features. (**b**) first 30 features.

For each layer, we extract the first 30 feature channels in descending order in terms of ratio to train the filter and obtain the convolutional object features per layer, as illustrated in Figure 4b. This paper obtains 3 new features by summing the first 30 features per layer, respectively. Then, a rich information feature can be obtained by summing the features of the three layers by utilizing different weights. The fused feature Fea_{new} is depicted in Figure 3f, and is written as:

$$Fea_{new} = \xi_1 Fea_{conv2-3} + \xi_2 Fea_{conv3-4} + \xi_3 Fea_{conv4-4} \tag{4}$$

where $Fea_{conv2-3}$, $Fea_{conv3-4}$, $Fea_{conv4-4}$ denote the depth features extracted by theconv2-3 layer, conv3-4 layer, conv4-4 layer, respectively, and $\xi_i(i = 1, 2, 3)$ is the weighting parameter.

This paper builds two independent appearance models by training the correlation filters separately using the fused features F_{CNN} and HOG features F_{HOG}. The decision power of the response is calculated using Average Peak Correlation Energy (*APCE*). We use the difference between the *APCE* values of two adjacent frames to decide the validity of the features. The average peak correlation energy (*APCE*) is calculated as:

$$APCE = \frac{|R_{max} - R_{min}|^2}{mean\left(\sum_{m,n}\left(R_{(i,j)} - R_{min}\right)^2\right)} \tag{5}$$

where R_{max} and R_{min} denote the maximum and minimum values of the feature response, respectively, and $R(i, j)$ denotes the response value at coordinate (i, j).

The decision-making power of different frames is represented by ϕ:

$$\begin{cases} \phi = \frac{1}{|APCE_t - APCE_{t-1} + \eta|} \ , \quad if\ APCE_t = APCE_{t-1} \\ \phi = \frac{1}{|APCE_t - APCE_{t-1}|} \ , \quad if\ APCE_t \neq APCE_{t-1} \end{cases} \tag{6}$$

where $APCE_t$ and $APCE_{t-1}$ are the *APCE* values of the feature response at frame t and $t-1$, respectively. The larger the ϕ value, the smaller the regional fluctuation of the adjacent feature responses and the stronger the validity of the corresponding feature. η is set to 0.01 to prevent the denominator from being zero.

The weights of the F_{HOG} and F_{CNN} are assigned according to the decision-making power. Thus, the weights of F_{HOG} can be expressed as:

$$W_{HOG} = \frac{\phi_{F_{HOG}}}{\phi_{F_{HOG}} + \phi_{F_{CNN}}} \tag{7}$$

where $\phi_{F_{HOG}}$ and $\phi_{F_{CNN}}$ denote the decision power values of the F_{HOG} and F_{CNN} features, respectively. Thus, the final features response is:

$$f(z) = W_{HOG} \times f_{HOG}(z) + (1 - W_{HOG}) \times f_{CNN}(z) \tag{8}$$

where f_{HOG} and f_{CNN} are the feature f_{HOG} and f_{CNN} response, respectively.

2.2. Subpixel Positioning Method

The final object position in the *KCF* algorithm is obtained by estimating the maximum value position of the correlation response patch. However, the maximum position calculation may incorporate errors that accumulate frame by frame as the object moves, imposing the tracking bounding box to drift or even losing the object. Thus, we develop a subpixel positioning method that obtains the maximum value position in the correlation response more accurately and detects the position of the peak value in the response patch. Our method's operating principle is that according to a set of discrete values, the coordinate m is the observed position of the extreme value $f(m)$ with left and right neighboring positions be $f(m-1)$ and $f(m+1)$, respectively, and the true extreme value is $m+\beta$ with β the estimated position offset.

The correlation response is illustrated in Figure 5, where P_3 is the peak response point and P_2, P_3, P_4, P_5 are the points around the peak point. For example, we consider the horizontal coordinate x of the peak position to illustrate how to determine a more accurate position. Assuming that the observed peak position in the current frame is x

with a response value $f(x)$ and the true peak position in the current frame is $x+\varepsilon$ with a response value $f(x+\varepsilon)$ where the ε is the offset value, Taylor's formula can approximate the peak point of the correlation response:

$$f(x+\varepsilon) = f(x) + f'(x)\varepsilon + \frac{1}{2}f''(x)\varepsilon^2 + \left(\varepsilon^3\right) \tag{9}$$

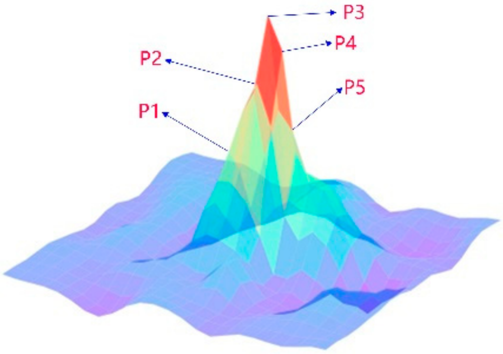


Figure 5. The graph of feature response.

The derivative at the peak of the continuous response is zero and is obtained by neglecting the highest order term, which can be expressed as

$$f'(x+\varepsilon) = f'(x) + f''(x)\varepsilon + \frac{1}{2}f'''(x)\varepsilon^2 \tag{10}$$

By solving the quadratic equations utilizing a small step size per move in adjacent frames, the solution of Equation (10) can be obtained as:

$$\varepsilon = \frac{-f''(x) + \sqrt{f''(x)^2 - 2f'''(x)f'(x)}}{f'''(x)} \tag{11}$$

$$f'(x) = \frac{f(x+1) - f(x-1)}{2} \tag{12}$$

$$f''(x) = \left[f'(x+1) - f'(x)\right] - \left[f'(x) - f'(x-1)\right] \tag{13}$$

$$f'''(x) = \left[f''(x+1) - f''(x)\right] - \left[f''(x) - f''(x-1)\right] \tag{14}$$

The true peak position can be written as

$$\begin{cases} x_{true} = x + \varepsilon_x \\ y_{true} = y + \varepsilon_y \end{cases} \tag{15}$$

where $\varepsilon_x, \varepsilon_y$ are the offset values in the x and y direction, respectively, and x_{true}, y_{true} are the true position of the peak.

In summary, the proposed method obtains an approximate offset value by calculating the peak position with subpixel positioning and utilizing the Taylor formula. This strategy reduces the object's drift and enables accurate object tracking.

2.3. Motion Trajectory Compensation

Correlation filter algorithms do not use the motion object state information for position prediction. In contrast, the KF optimally estimates the state of a stochastic dynamic system by combining information such as target velocity and acceleration to predict the position of the next frame, enabling accurate object position prediction under rapid motion, motion blur, and occlusion. The KF method is computationally low-cost, affording fast object tracking.

The KF provides the optimal signal estimation in the time domain utilizing a linear minimum variance estimation scheme. However, applying the filter requires setting the filter parameters, which are not trivial, during the system determination and noise measurement. Nevertheless, both noises hugely impact the filter′s estimation effectiveness. Hence, to ease the calculations, it is generally considered that both follow a normal distribution with zero mean and are constant throughout the time series. However, in the actual KF, the state and measurement system noise variances are set empirically, with a certain degree of randomness and blindness. After performing a relevant case study, we conclude that different variance values heavily impact the filtering results. The system′s noise cannot be eliminated, and thus it is essential to reduce its interference as the noise variance is directly related to the extent that noise impacts the KF results. Next, we discuss the effect of variance on KF.

KF is a real-time recursive algorithm that utilizes the system′s estimate (state or parameter) originating from the filter′s output and estimates the system to be processed based on the system and the observation equations. Essentially, KF is an optimal estimation method, with its mathematical model presented next. Let the state and the measurement equations of the stochastic linear discrete system be denoted as:

$$\begin{cases} X_t = A_{t,t-1}X_{t-1} + \omega_{t-1} \\ Z_t = H_t X_t + v_t \end{cases} \tag{16}$$

where X_t and Z_t are the system′s state and observation vectors at time t, respectively, $A_{t,t-1}$ is the state transfer matrix, H_t is the observation matrix, ω_{t-1} is the process noise matrix, and v_t is the observation noise matrix.

The prediction and update equations for the Kalman filter are:

$$\begin{cases} X_{t,t-1} = A_{t,t-1}X_{t-1} \\ P_{t,t-1} = A_{t,t-1}P_{t-1}A_{t,t-1}^T + Q_{t-1} \\ K_t = P_{t,t-1}H_t^T\left(H_t P_{t,t-1}H_t^T + R_t\right)^{-1} \\ X_t = X_{t,t-1} + K_t(Z_t - H_t X_{t,t-1}) \\ P_t = (I_t - K_t H_t)P_{t,t-1} \end{cases} \tag{17}$$

where $X_{t,t-1}$ is the state prediction at time t, X_{t-1} is the optimal estimate at time $t-1$, $P_{t,t-1}$ is the state covariance matrix at time t, Q_{t-1} is the noise covariance matrix of ω_{t-1}, R_t is the covariance matrix of the measurement noise v_t, K_t is the gain matrix, and Z_t is the input variable matrix at time t.

From Equations (33) and (34), we observe that the gain matrix K_t is related to the initialization P_0, the system process noise covariance matrix Q_t, and the measurement noise covariance matrix R_t. Moreover, the gain matrix decreases as R_t increases because if the measurement noise increases, it causes a significant error, and therefore, the filter gain should be set to a smaller value to reduce the effect of the observation noise on the filter value. If P_0 and Q_t become smaller, the process noise covariance matrix $P_{t,t-1}$ and the optimal filter covariance matrix P_t become smaller, and the gain matrix K_t decreases. Therefore, the gain matrix K_t is proportional to Q_t and inversely proportional to R_t. K reflects how close the filter value is to the actual value. The larger the K, the greater the difference between the filter and the actual value, and vice versa.

The state and measurement equation of the KF can be easily determined when the system is stable. However, in reality, the system process and measurement noise are unknown and thus are generally treated as white noise. In fact, the system process and measurement noise are dynamic. Thus, to simplify the calculations, the system process and measurement noise covariances are set as constants, where the specific values are heuristically set, involving some blindness and thus may reduce the filtering effect. Spurred by this, we improve the KF by improving the convergence speed and altering the covariance values. The related process initially calculates the *SDM* value of the adjacent frame object to decide whether to adjust the process noise covariance of the KF, which can be written as:

$$SDM = \zeta \times (1/V_{max}) + (1-\zeta) \times d \tag{18}$$

$$d = \sqrt{(C_r - C_k)^2} \tag{19}$$

where C_r is the object center position of the current frame obtained by the *KCF* tracker, C_k is the object center position of the current frame obtained by the *AKF* tracker, V_{max} is the maximum feature response, d is the Euclidean distance, and ζ is a parameter set to 0.4.

When the *SDM* value is less than the occlusion threshold (*OT*), i.e., the object is not occluded, we combine *AKF* and *KCF* trackers to track the object and obtain its position in the current frame. Simultaneously, the estimated position is further exploited to select the search region in the next frame, avoiding searching in the wrong region. The final object position (*FOP*) is:

$$\begin{cases} FOP = (1-\gamma)OP_{KCF} + \gamma OP_{AKF} \ , \quad SDM \leq OT \\ FOP = OP_{AKF} \ , \quad SDM > OT \\ FOP = OP_{KCF} \ , \quad without\ convergence \end{cases} \tag{20}$$

where OP_{KCF}, OP_{AKF} are the object center positions calculated from the *KCF* and *AKF* trackers, respectively, and *OT* is the occlusion threshold that determines whether the object is occluded or not. After several experimental verifications, we set the occlusion threshold (*OT*) to 3.2. When the *SDM* value exceeds 3.2, the object is heavily occluded or disappears. The selection of the occlusion threshold will be described in detail in Section 4.2.

The algorithm presented in this paper constructs an *AKF* tracker utilizing the object position obtained by the correlation filter as the observation value to compensate for the object motion trajectory and improve tracking accuracy. During the *AKF* tracker operation, two critical issues need to be addressed. The first is the convergence problem, where judging the *AKF* tracker's convergence determines whether the object's motion trajectory can be compensated for the tracking result. This paper solves the convergence problem by calculating the Euclidean distance between the object center position obtained by the *KCF* and the *AKF* trackers, which determines whether the *AKF* tracker has reached convergence, expressed as:

$$\sqrt{OP_{KCF} - OP_{AKF}} \leq 2 \tag{21}$$

when the Euclidean distance is less than 2 pixels in 5 consecutive frames, we consider that the *AKF* tracker has converged.

Another problem is setting the state variables. The object's system state X_t is $X_t = [x_{center}, y_{cnter}, \ \Delta V_x, \Delta V_y]$, where x_{center}, y_{cnter} are the center position's horizontal and vertical object coordinates per frame, $\Delta V_x, \Delta V_y$ are the object's speed in the horizontal and vertical coordinate directions, respectively, and $Z_t = [x_{center}, y_{cnter}]$ is the object measurement value, i.e., the object's center position. Since the object's motion time in adjacent frames is very short, the object's motion is regarded as a short-time uniform linear motion, and thus we set the transfer matrix A and observation matrix H to be:

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{22}$$

The noise covariance matrix R and Q in original Kalman filter are:

$$R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad Q = \begin{bmatrix} 0.01 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0.01 \end{bmatrix} \tag{23}$$

when the SDM value exceeds OT, the object may be occluded. The process noise covariance matrix R_{update} and the observation noise covariance matrix Q_{update} can be written as:

$$R_{update} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, \quad Q_{update} = \begin{bmatrix} 0.001 & 0 & 0 & 0 \\ 0 & 0.001 & 0 & 0 \\ 0 & 0 & 0.001 & 0 \\ 0 & 0 & 0 & 0.001 \end{bmatrix} \tag{24}$$

This means that the object is occluded and the SDM value is greater than OT, so the noise covariance R_{update} should be reduced and Q_{update} should be increased to afford the adaptive Kalman filter predicting the position of the occluded object more accurately. If the SDM value is less than or equal to OT, R_{update}, Q_{update} can be expressed as:

$$\begin{gathered} R_{update} = \begin{bmatrix} SDM & 0 \\ 0 & SDM \end{bmatrix} \\ Q_{update} = \begin{bmatrix} (1-SDM)^2 & 0 & 0 & 0 \\ 0 & (1-SDM)^2 & 0 & 0 \\ 0 & 0 & (1-SDM)^2 & 0 \\ 0 & 0 & 0 & (1-SDM)^2 \end{bmatrix} \end{gathered} \tag{25}$$

The object position is predicted from the second frame onwards, and the position of the target is calculated by Equation (24), which is then input into the tracker to continue tracking the target until the end of the tracking process.

It should be noted that KF suffers from a convergence problem. If KF does not converge once occlusion appears, the object is lost, and the tracking fails. Table 1 highlights that KF achieves convergence at 32 frames on average, while AKF at 15 frames on average, compensating and correcting the object trajectory as soon as possible, thus reducing the tracking error and improving the performance and robustness of the object tracking process.

Table 1. Convergence frames of two filters in eight video sequences. The bold values denote the average frame numbers on all video sequences.

	Plane1	Plane2	Car1	Car2	Car3	Car4	Car5	Car6	Average
KF	35	32	33	29	30	31	28	34	**32**
AKF	15	14	16	13	15	14	13	17	**15**

2.4. Solution for Object Occlusion

The correlation filter algorithms track the object quickly and accurately under normal circumstances, but when the object obstacles block the object, the object tracking fails if the area of the tracked object is reduced or even disappears temporarily. The problem of object occlusion has constantly challenged object tracking, and therefore studying and solving this problem has theoretical and practical significance. Objects in the satellite videos are very small, occupying only a few pixels, so their complete occlusion is expected. As illustrated in Figure 6, when the car passes under the bridge, it becomes occluded and eventually disappears, causing most trackers to lose the object and struggle to re-track it when it re-appears. Therefore, object occlusion has always been challenging for computer vision applications.

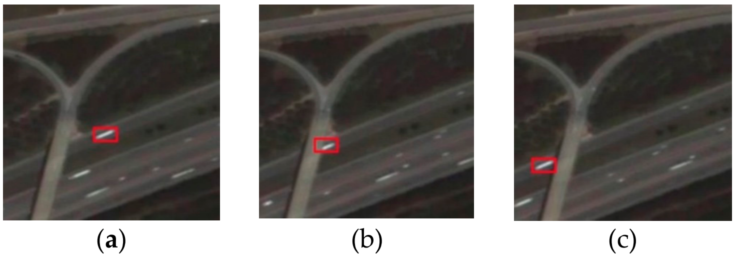


Figure 6. Visualization of an object occlusion process. (**a**) not occluded object. (**b**) partially occluded object. (**c**) end of object occlusion.

Hence, the following sub-problems need to be solved correctly to track the occluded object.

(1) *Occlusion Detection*: The algorithm must detect that the object is occluded (Figure 6a).

(2) *Occlusion Processing*: When the object is completely or partially occluded (Figure 6b), the algorithm must track the object to ensure that the occluded object is not lost.

(3) *End of Occlusion Detection*: The algorithm must be able to detect the end of the occluded object and correctly re-track the object when it re-appears (Figure 6c).

To solve the above problems, our algorithm involves the following steps:

(1) Update the tracking model only when the confidence of the object tracking is relatively high to avoid the object model from being degraded and simultaneously increase the processing speed. We use the SDM value of the correlation response patch to judge the tracking quality. The smaller the SDM value, the better the tracking result. Object tracking has several significant difficulties: object deformation, illumination variation, the object's blur motion, the fast motion of the object, background clutter, object rotation, scale variation, and object occlusion. These reduce the SDM value of the relevant response patch. However, illumination variation, scale variation, and motion blur are not evident in satellite videos and thus can be neglected in object tracking. The primary reason for a higher SDM value in the correlation response patch is the object's partial or complete occlusion. In summary, the SDM value can be used to judge whether the object is occluded or not, and choosing the appropriate threshold is the key to solving the object occlusion problem.

(2) When the object is occluded, we avoid erroneous features from interfering with the object features by stopping updating the tracker's filter. Moreover, the position calculated by the KCF tracker is not accurate and cannot be used as the object position of the current frame. Therefore, it is necessary to utilize the KF algorithm to predict the object's position.

(3) We compare the SDM value with the occlusion threshold. If the SDM value is less than the occlusion threshold, the object is not occluded and can be tracked normally. Otherwise, the object is partially or entirely occluded, and KF is exploited to track the occluded object.

Using KF to solve the object occlusion problem when the object re-appears allows to relocate it precisely. The corresponding pseudocode of the proposed improved method is presented in Algorithm 1.

3. Experiments

This section evaluates our algorithm on eight satellite videos, sets the relevant parameters, examines its tracking effect utilizing evaluation metrics, and challenges its performance against several classic algorithms.

3.1. Video Datasets and Compared Algorithms

The datasets employed are captured by the Jilin-1 satellite constellation. The satellite video sequences are all cropped from the original video sequences, and most of them contain three types of moving objects: moving vehicles, airplanes in the sky, and ships in the ocean. There are three airport objects, two moving vehicle objects, and ships in the ocean. The total number of objects moving in the airport is approximately 10. The maximum size of the object in all satellite videos is 32 × 27 pixels, and the minimum is about 10 × 10 pixels. We number every frame and select the object in a frame as a representative to zoom in to see it more clearly.

Algorithm 1 The proposed tracking scheme

```
Input:
  frames: video datasets. $T$: number of processed frames.
  $F_T$: current frame $T$. $P_{T-1}$: the object position of frame $T-1$.
Output:
  $P_T$: the current frame object position.
Selcct the region of interest (ROI) and set the position of first frame.
Set the occlusion threshold $T_h$.
for i in range (len(frames)):
    if i == 1: (first frame)
      Initialize the KCF tracker, VGG network, and Kalman filter.
      $F_{HOG}$: extract HOG features. $F_{VGG}$: VGG features selection and enhancement.
      $P_T$: the position of current frame.
    else:
      Crop image patch from frames [i] according to $\mathbf{P_T}$.
      Fuse-response: Fusion strategy for feature ($F_{HOG}$, $F_{VGG}$) responses.
      $P_{peak}$: the position of the max fuse-response.
      SDM: Calculate the SDM to detect occlusion.
      if SDM > $T_h$:
        /* the object is unoccluded */
        $P_{sub\text{-}peak}$: Subpixel location for $P_{peak}$.
        $P_{final}$: Motion trajectory compensation and correction.
      else:
        $P_{final}$: The object position obtained by Kalman filter.
        $P_T \leftarrow P_{final}$
      return $P_T$
break
```

We challenge our algorithm against the current object trackers: CSK [11], KCF [12], HCF [26], TLD [27], DSST [28], SiamFC [29], SiamRPN [30]. The CSK algorithm and KCF algorithm are improved based on the MOSSE algorithm. The TLD algorithm has a better effect on the tracker when the object is partially occluded, while the HCF, SiamFC, and SiamRPN algorithms all extract the object's depth features during tracking, achieving appealing tracking results. The subsequent trials highlight that our proposed method is more effective for object tracking in satellite videos than current classic algorithms.

3.2. Parameters Setting

The proposed algorithm is developed in Python, while KCF, CSK, TLD algorithms are implemented through the Open CV API. All tracking methods are implemented on a 3.5-GHz Intel Xeon E3 1240 v5 CPU and an NVIDIA GeForce GTX 1080Ti GPU. The KCF and our algorithm utilize the HOG features with a cell size of 4 × 4. The search window size in the expanded object area is 2.5 times the original area. Additionally, the regularization parameter λ is set to 0.0001, the learning rate θ is 0.012, the bandwidth of the Gaussian kernel function σ_1 is 0.6, and the bandwidth of the 2-D Gaussian function is $\sqrt{wh}/16$, where w and h are the width and height of the object bounding box, respectively. The remaining parameters are the ones originally proposed, while ς_1, ς_2, ς_3 are the 1, 0.6, 0.2 of Equation (17), respectively.

3.3. Evaluation Metrics

For a fair evaluation, this paper uses two metrics as quantitative analysis indicators: tracking success rate and tracking precision [31,32].

We use the center location error (CLE) to evaluate tracking precision, i.e., the Euclidean distance between the object's predicted and real center position. CLE is calculated by:

$$CLE = \sqrt{\left(x_p - x_{gt}\right)^2 + \left(y_p - y_{gt}\right)^2} \tag{26}$$

where $\left(x_p, y_p\right)$ is the object center position predicted by our proposed algorithm, $\left(x_{gt}, y_{gt}\right)$ is the object's ground truth location. The tracking precision is the ratio of the frames whose center position error obtained by the tracking algorithm is less than a certain threshold to the total number of the video frames. Therefore, the threshold has a significant influence on the object tracking precision. Since an object in the satellite videos is very small, we set the threshold to five, i.e., if the CLE value is within five pixels, we consider that the object is successfully tracked.

The tracking algorithm's success rate is the ratio of the frames whose overlap rate exceeds a certain threshold to the total number of video frames. Accordingly, the overlap rate is the ratio of the overlap area between the predicted and the real object tracking frame to the total area of both areas, expressed as:

$$S = \frac{Area\left|r_p \cap r_{gt}\right|}{Area\left|r_p \cup r_{gt}\right|} \tag{27}$$

where r_p and r_{gt} are the predicted bounding and the real object tracking area, $\cap$ and $\cup$ are the intersection and union, respectively, and $||$ is the number of pixels in the region.

The threshold can be set to 0.5. If the overlap ratio of the current frame is greater than 0.5, we consider that the object tracking of the current frame is successful. Therefore, we evaluate the overall performance of our proposed algorithm on all satellite videos utilizing the precision plot, success plot, and area under curve (AUC). Moreover, the success score, precision score, and AUC can further rank the trackers.

Moreover, the FPS denotes the number of frames that the tracker can process, so it is suitable to evaluate the tracking speed per tracker.

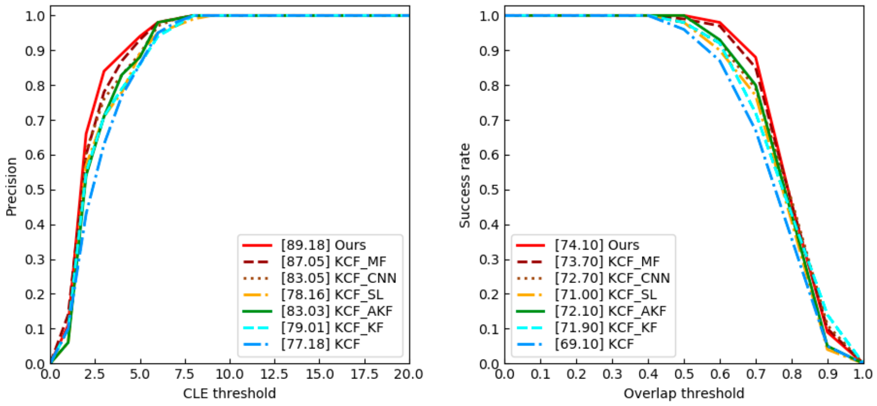
4. Results and Analysis

4.1. Ablation Study

Our tracker involves three critical modules: multi-feature response fusion, subpixel localization, and motion trajectory correction and compensation. To evaluate their separate impact on the tracker's performance, we implement several variations of our tracker investigating the contribution of each module.

The KCF tracker solely utilizing the multi-feature fusion strategy is denoted as KCF_MF, with CNN features as KCF_CNN, with subpixel localization as KCF_SL, with the adaptive Kalman filter as KCF_AKF, and with only the Kalman filter as KCF_KF. Finally, the proposed algorithm is denoted as Ours. The corresponding precision and success plots are illustrated in Figure 7 and Table 2, highlighting that the corresponding scores of the KCF_SL tracker are increased by 1% and 1.4%, respectively, compared to the KCF tracker. Although the scores are not significantly increased, the subpixel localization method manages to locate the object and improve the tracking performance accurately. The KCF_KF and KCF_AKF tracker increase the precision score by 5.8% and 1.8%, respectively, and the success score by 3.3% and 2.1%.

Combining the KCF tracker and Kalman filter assists the tracker in correcting the tracking results and reduces the risk of drifting because the filter converges earlier and reduces tracking errors in time. The KCF_CNN and KCF_MF trackers also attain appealing tracking results increasing the precision score by 9.9% and 5.9%, respectively, and the success score by 6% and 4.4%. This performance enhancement is because the KCF_CNN tracker extracts the object's depth features, achieving appealing tracking results. Moreover, the KCF_MF tracker combines the HOG features and depth features to enhance the object's description further; thus, extracting multiple features from the object can improve tracking accuracy.



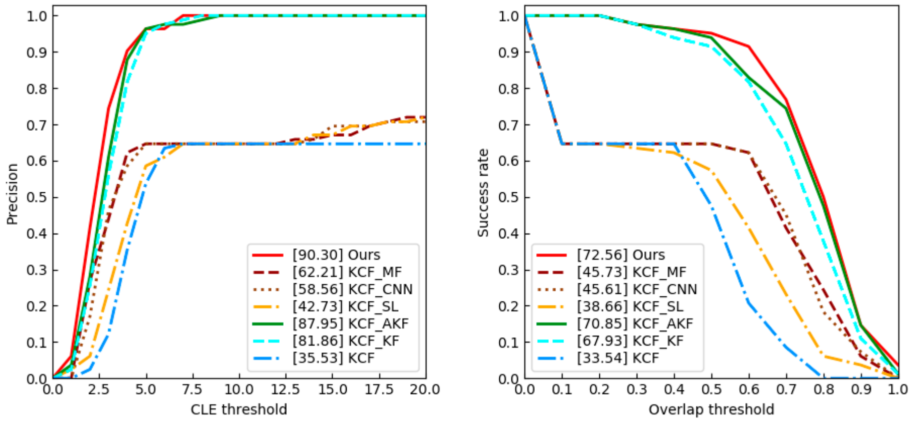
(**a**) (**b**)

Figure 7. Ablation study on all the unoccluded video sequences. The legend in the precision and success plot are the precision and AUC value score per object tracker, respectively. (**a**) Precision plots; (**b**) Success plots.

Table 2. The experimental ablation results of all un-occluded video sequences (in bold the optimal results per tracker).

	Ours	**KCF_MF**	**KCF_CNN**	**KCF_SL**	**KCF_AKF**	**KCF_KF**	**KCF**
AUC (%)	**74.1**	73.7	72.7	71.0	72.1	71.9	69.1
Precision score (%)	**89.2**	87.1	83.1	78.2	83.0	79.0	77.2
Success score (%)	**97.2**	96.4	94.8	91.8	93.7	92.5	90.4
FPS	18	22	20	94	92	93	**96**

Our tracker embedded with all three modules makes the tracker highly robust. The precision and success plots on the occluded videos are presented in Figure 8 and Table 3, highlighting that only our algorithm, KCF_AKF, and KCF_KF can solve the object occlusion problem, while the remaining algorithms perform poorly. This shows that both the adaptive Kalman filter and Kalman filter afford predicting the occluded object's position and continue to track it when the object occlusion ends.



(**a**) (**b**)

Figure 8. Ablation study on all the occluded video sequences. The legend in the precision and success plot are the precision and AUC value score per object tracker, respectively. (**a**) Precision plots; (**b**) Success plots.

Table 3. The experimental ablation results of all occluded video sequences (in bold the optimal results per tracker).

	Ours	**KCF_MF**	**KCF_CNN**	**KCF_SL**	**KCF_AKF**	**KCF_KF**	**KCF**
AUC (%)	**72.6**	45.7	45.6	38.7	70.9	67.9	33.5
Precision score (%)	**90.3**	62.2	58.6	42.7	88.0	81.9	35.5
Success score (%)	**95.1**	64.8	64.6	58.4	93.9	92.5	47.6
FPS	17	20	21	95	92	93	**97**

Based on the above analysis and tracking results, the three crucial modules proposed in this paper improve the object tracking performance. Our method can also be applied to other trackers for object tracking on the ground and obtain improved results.

4.2. Object Occlusion Analysis

The appropriate threshold can help us judge whether the object is occluded or not. Therefore, how to select the appropriate threshold is very important. This paper selects the appropriate threshold by the SDM value of the response patch of all satellite video sequences. The distribution of the SDM value is unimodal in Figure 9. If the SDM value exceeds the occlusion threshold in five consecutive frames, the object can be considered partially or entirely occluded. Therefore, we can utilize the SDM value to judge whether the object is occluded or not.

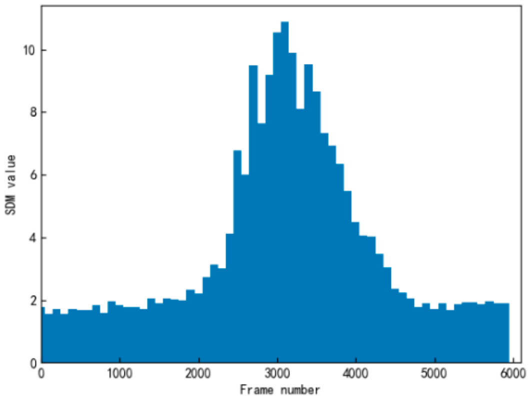


Figure 9. Distribution of the SDM value of the response patch. We can see an obvious unimodal distribution. Hence, selecting an appropriate threshold to judge occlusion and non-occlusion.

In order to select the appropriate threshold, we utilize the grid search from [2.0, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 3.0, 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.8, 3.9, 4.0]. The success plot for per threshold is shown in Figure 10. We can see that when the threshold is 3.2, the AUC is the maximum. Therefore, 3.2 can be selected as the occlusion threshold.

The visualization of the occlusion process in Figure 11 illustrates the relationship between the object occlusion's state and the SDM value. The threshold 3.2 can well judge whether the object is occluded. The SDM value exceeds the threshold when the object is partially or completely occluded. When the object is not occluded, the SDM value is less than the threshold in satellite videos. Some extreme cases may exist where the SDM value above the threshold is not caused by the object occlusion but caused by illumination variation or motion blur of the object. Since the duration of those extreme cases is very short, the impact on the final tracking results is negligible. Hence, our improvements are suitable not only for occluded objects but also for non-occluded objects.

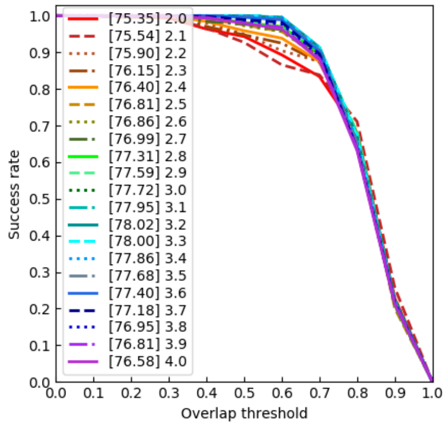
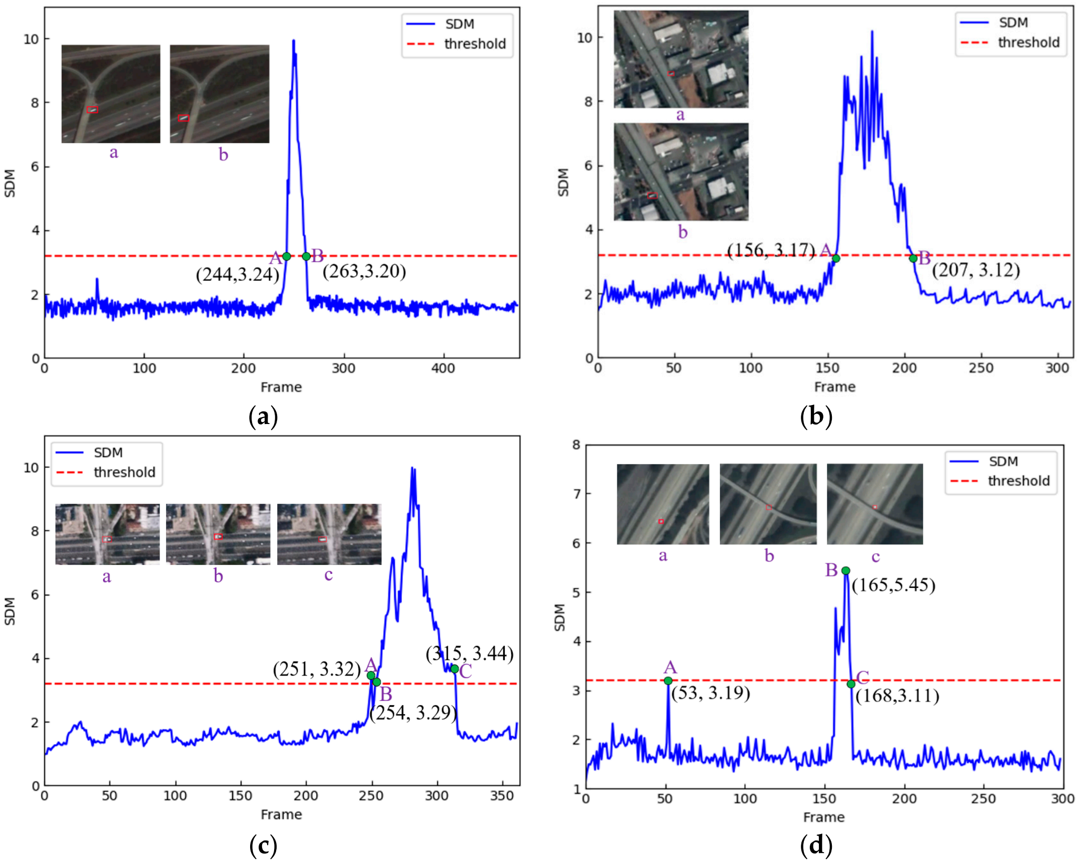


Figure 10. Success plot for per threshold and the legend is the AUC per threshold.



(**a**)

(**b**)

(**c**)

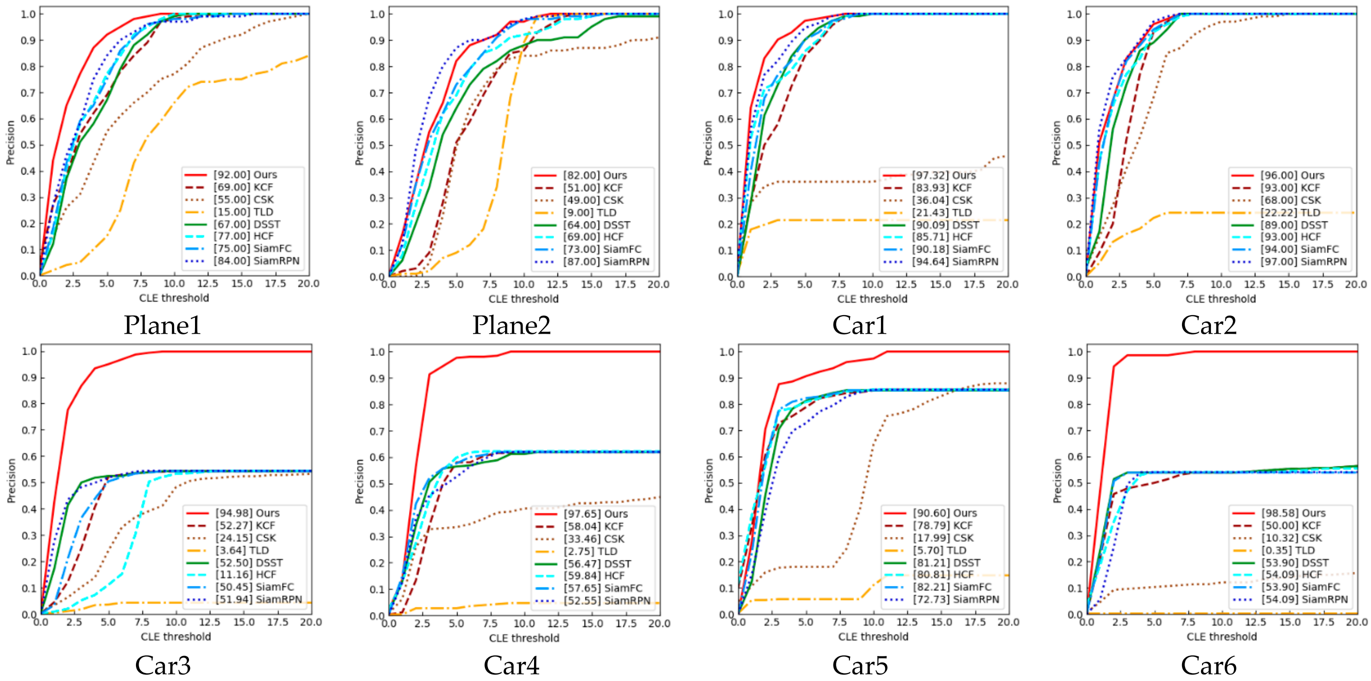
(**d**)

Figure 11. Visualization of some tracking results for the occluded object. The data in the parenthesis marked by upper case letters denote the current frame's SDM value of the images with the corresponding lower case letters. (**a**) Occlusion process of Car3 sequences. (**b**) Occlusion process of Car4 sequences. (**c**) Occlusion process of Car5 sequences. (**d**) Occlusion process of Car6 sequences.

4.3. Tracking Result Analysis

4.3.1. Quantitative Evaluation

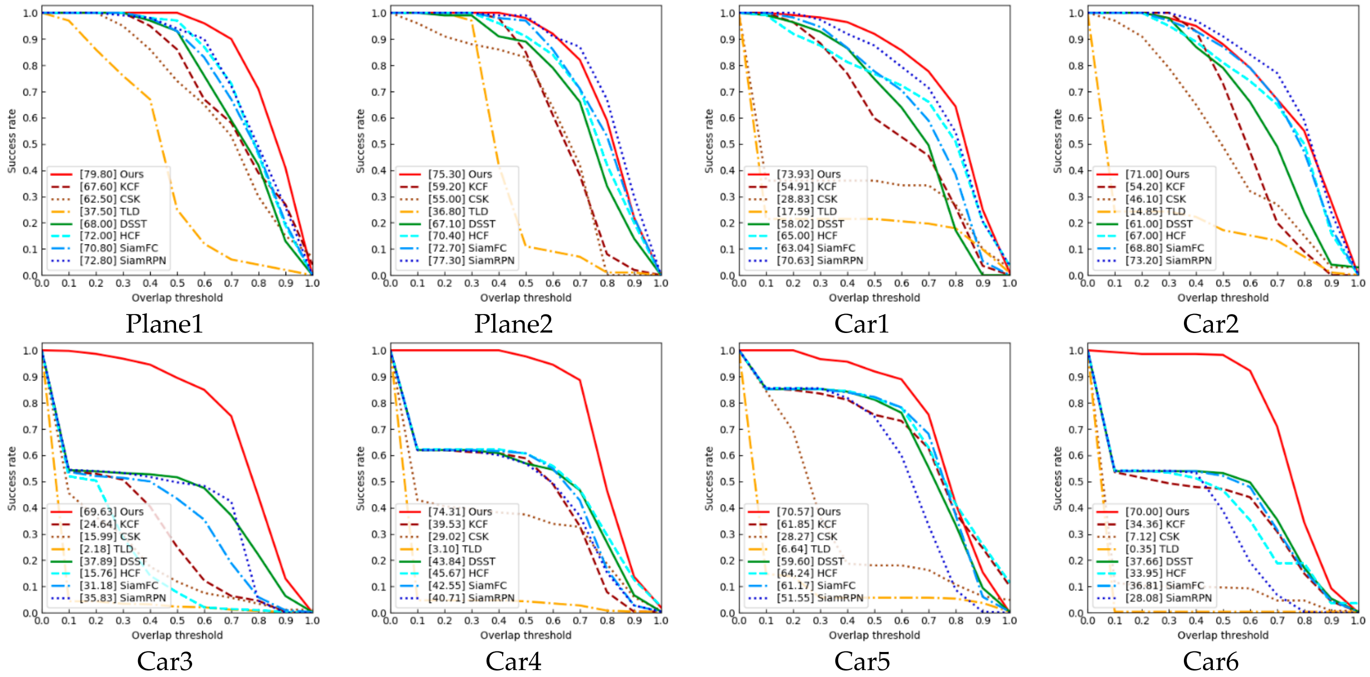
Figures 12 and 13 illustrate our algorithm's precision and success plots against other classical methods on eight satellite videos. The proposed method achieves the optimal tracking results in the most challenging scenarios. Compared with KCF, the suggested tracker improves tracking accuracy by utilizing CNN features and trajectory compensation correction. Given that the original KCF tracking accuracy is low and the results are not appealing, modifying KCF to handle various situations can significantly improve the object tracking performance.



Plane1 Plane2 Car1 Car2

Car3 Car4 Car5 Car6

Figure 12. Precision plots of eight video sequences involving object sequences without and with occlusion. The legend in the precision plot is the corresponding precision score per object tracker.



Plane1 Plane2 Car1 Car2

Car3 Car4 Car5 Car6

Figure 13. Success plots of eight video sequences involving object sequences without and with occlusion. The legend in the success plot presents the corresponding AUC per object tracker.

To analyze our algorithm's performance, we divide the satellite videos into two parts: four video sequences without object occlusion and four video sequences with object occlusion. The tracking results of the unoccluded object on the Plane1, Plane2, Car1, and

Car2 sequences are presented in Figure 12 and Table 4. Regarding the Plane1 and Car1 sequences, our algorithm ranks first and attains better results on the AUC, success score, and precision score metrics. Compared to the SiamRPN tracker, the proposed method improves the AUC, success score, and precision score by approximately 7%, 6%, and 2.3% for Plane1 and 3.3%, 4.1%, and 2.7% for the Car1 object, respectively. The SiamRPN tracker combines the region proposal network (RPN) with a Siamese network and directly classifies and regresses the position of K anchor points on different scales and aspect rations at the same location, improving the object tracking accuracy. However, on the Car2 and Plane2 sequences, the SiamRPN algorithm ranks first. Nevertheless, our algorithm's AUC, success score, and precision score are increased by approximately 2%, 1%, and 5%, respectively, for the Plane2 object and 4.2%, 3%, and 1% for the Car2. Thus, the developed algorithm does not differ substantially from the SiamRPN tracker in terms of performance and robustness.

Table 4. The object tracking results of all video sequences. Employing AUC, success score, and precision score to determine the best tracker. The bold-digital denotes the optimal results.

Video Datasets	Evaluation Metrics	Ours	KCF	CSK	TLD	DSST	HCF	SiamFC	SiamRPN
Plane1	AUC (%)	**79.8**	67.6	62.5	37.5	68.0	72.0	70.8	72.8
	Precision score (%)	**99.0**	88.6	77.4	30.2	94.3	97.3	94.6	96.7
	Success score(%)	**100.0**	86.0	74.0	25.0	93.0	97.0	93.0	94.0
	FPS	14	95	**106**	13	76	21	16	13
Plane2	AUC(%)	75.3	59.2	55.0	36.8	67.1	70.4	72.7	**77.3**
	Precision score(%)	82.0	51.0	49.0	9.0	64.0	69.0	73.0	**87.0**
	Success score(%)	98.0	85.0	83.0	11.0	89.0	91.0	97.0	**99.0**
	FPS	15	90	**91**	21	86	34	12	9
Car1	AUC(%)	**73.9**	54.9	28.8	17.6	58.0	65.0	63.0	70.6
	Precision score(%)	**97.3**	83.9	36.0	21.4	90.1	85.7	90.2	94.6
	Success score(%)	**91.6**	59.8	36.0	21.4	74.8	76.8	77.7	87.5
	FPS	22	89	**97**	19	63	28	20	19
Car2	AUC(%)	71.0	54.2	46.5	0.7	60.3	65.9	67.8	**75.2**
	Precision score(%)	89.0	77.0	54.0	1.0	83.0	84.0	87.0	**90.0**
	Success score(%)	88.0	73.0	49.0	17.2	79.0	81.0	87.0	**91.0**
	FPS	14	105	**112**	23	95	19	13	10
Car3	AUC(%)	**69.6**	26.9	18.6	0.2	40.9	16.7	34.9	39.9
	Precision score(%)	**86.3**	21.4	8.2	0.2	49.1	0.7	33.6	52.6
	Success score(%)	**89.5**	25.0	11.8	2.3	51.6	7.7	43.4	49.7
	FPS	26	78	**84**	35	62	22	20	15
Car4	AUC(%)	**74.3**	40.2	30.3	0.4	45.6	45.7	43.7	54.3
	Precision score(%)	**86.3**	36.9	35.8	0.4	54.5	42.9	54.5	68.7
	Success score(%)	**97.6**	58.8	37.4	4.3	56.9	60.6	60.8	56.9
	FPS	22	**89**	82	33	80	22	18	15
Car5	AUC(%)	**70.6**	66.5	47.1	0.3	62.6	67.1	62.6	55.8
	Precision score(%)	**85.2**	77.1	34.6	0.3	68.8	73.7	75.5	67.0
	Success score(%)	**91.9**	75.4	18.0	5.7	81.2	81.8	82.2	74.7
	FPS	24	**114**	108	29	100	33	21	17
Car6	AUC(%)	**70.0**	36.7	9.6	2.9	40.2	36.5	39.3	30.6
	Precision score(%)	**90.8**	41.8	8.2	0.4	47.9	32.4	47.9	32.0
	Success score(%)	**98.2**	47.2	9.6	0.3	53.2	46.6	52.1	38.8
	FPS	28	89	**94**	24	81	23	16	12

Overall, the SiamRPN algorithm performs poorer than our method because it is trained offline, the model parameters are not updated online, and the number of anchor points at the same position should be designed to be large enough. The latter undoubtedly increases the number of network parameters and generates a large number of negative samples, resulting in unbalanced positive and negative samples, slightly lowering the tracking accuracy of SiamRPN compared to the suggested algorithm. Another reason is that when the target rotates, the traditional object tracking algorithms cannot resist this change and still update the original trajectory, generating errors and causing the object tracking bounding box to drift, significantly reducing tracking accuracy. The SiamRPN algorithm uses multiple anchor points and scales to overcome these changes and achieve appealing tracking results.

The remaining four video sequences (Car3, Car4, Car5, and Car6) involve occluded object sequences. Many classical tracking algorithms cannot continue tracking the object when the moving object is occluded, while only our algorithm still tracks the object. This may be because the classical tracking algorithms cannot deal with long-time object occlusion and thus track the wrong object, eventually losing the object. In contrast, our algorithm combines the Kalman filter with occlusion detection to determine whether the object is occluded. When the object is occluded, we no longer utilize the KCF algorithm to track the occluded object and employ the adaptive Kalman filter to track the occluded object by manually setting the noise covariance Q and R values, which are initially set to 2 and 0.01, respectively. The precision score, success score, and AUC of various tracking algorithms are illustrated in Figures 12 and 13 and Table 4. The analysis of the above observations indicates that since our algorithm can solve the problem of occluded objects, it can track the object when it re-appears. This is why it has a high tracking accuracy, while the competitor algorithms have lower tracking results and lose the object due to occlusion.

In summary, objects in satellite videos easily encounter occlusion due to their small size and large image width, with the object occlusion problem being a significant difficulty in object tracking. Thus, solving the object occlusion problem is the key to successful tracking, while for the unoccluded case, the object is tracked according to the original method.

4.3.2. Qualitative Evaluation

This section chooses three representative trackers for qualitative evaluations against our proposed method, as presented in Figure 14, where the tracked objects are planes and vehicles. Figure 14 indicates that the selected four trackers show different tracking results on the Plane1, Plane2, Car1, and Car2 sequences, where all objects are unoccluded. On the Plane1 and Plane2 sequences, our method and the SiamRPN tracker achieve good tracking results. In frame 34 (Plane1) and frame 53 (Plane2), we observe that the selected four trackers can accurately track the object in Figure 14, while in frame 89 (Plane1) and frame 82 (Plane2), the KCF and SiamFC tracker present some tracking drift but can still track the object. As the object continues to move, our method and the SiamRPN tracker continue to track the object for the rest of the moving process, while the other two trackers present a significant drift and eventually lose the object. In frame 213 (Plane1) and frame 251 (Plane2), the object in the Plane2 sequences has a complex background with some similar interferences in the surroundings, increasing the tracking difficulties.

Our method has similar interferences in the surroundings, increasing the tracking difficulties. Our method can track the object because we extract multiple objects' features and then compensate and correct the object's motion trajectory. The objects in the Car1 and Car2 sequences are both moving vehicles. Regarding the Car2 sequences, due to the change of the object motion direction, all trackers except for SiamRPN suffer from bounding box drift, and thus SiamRPN can continue to track the object accurately, probably because SiamRPN employs the multi anchor points method, which can resist the object from short rotation within a short period.

Considering the Car3, Car4, Car5, and Car6 sequences, all objects are occluded by the bridge, and our tracker is the only one that continues tracking the object without losing it. When the object is occluded, the adaptive Kalman filter in our method predicts the object position in the next frame by automatically increasing the covariance noise R and decreasing the covariance Q. When the target occlusion ends, our method can immediately re-track the target. Opposing, the other trackers cannot handle the occlusion problem and are unable to predict the object's position, so when the target is occluded, the target is lost, resulting in tracking failure.

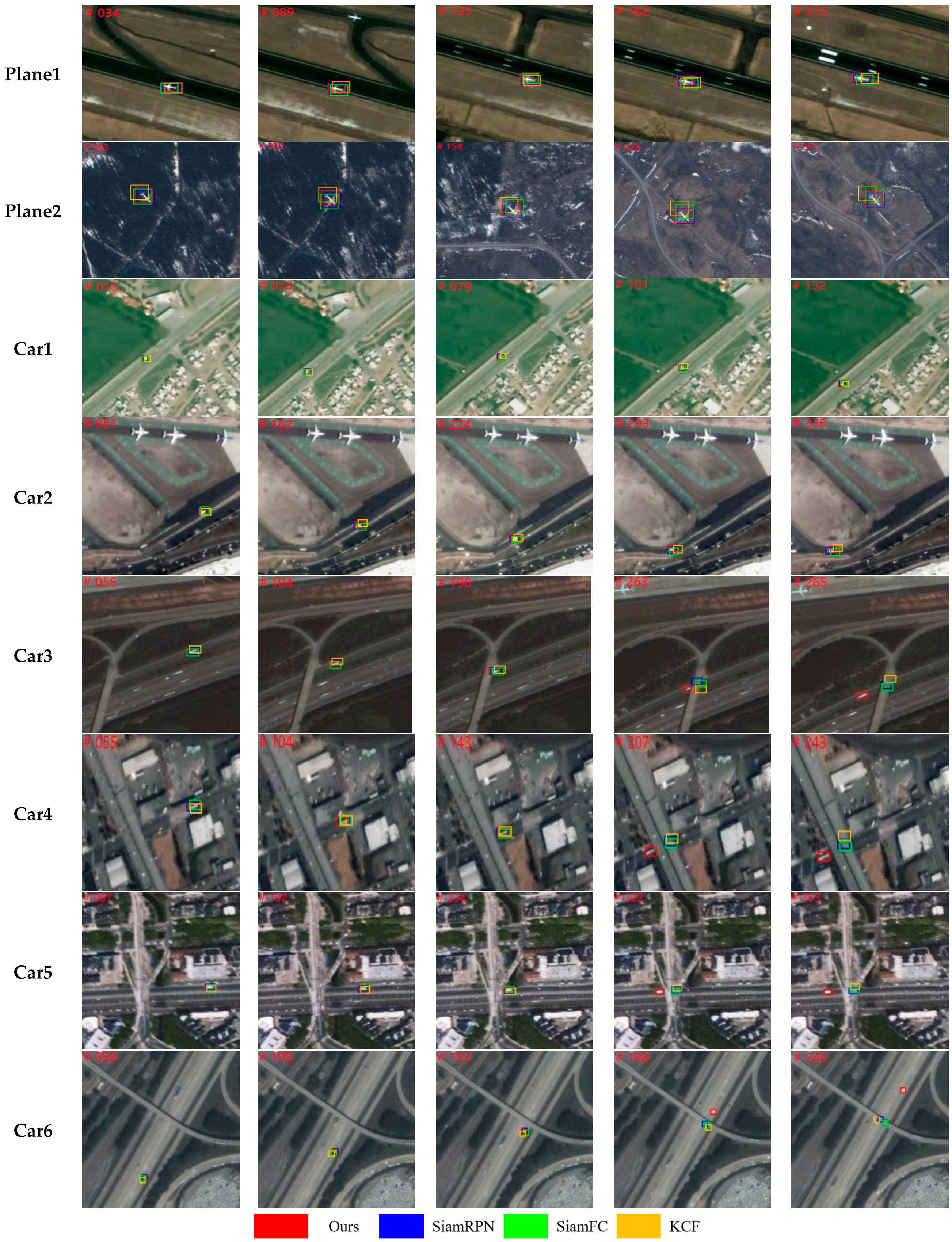


Figure 14. Screenshots of some tracking results without occlusion. At each frame, the bounding boxes with different colors are the tracking results of the different trackers and the red number in the top-left corner is the frame number of the current frame in the satellite videos.

In conclusion, the visual tracking results show that our method can solve the problem of short-time target occlusion and is suitable for satellite video targets with complex backgrounds and fast movements, improving the object's tracking accuracy.

5. Conclusions

We design a novel tracking algorithm that improves the robustness and tracking accuracy of object tracking in satellite video under complex background and object occlusion. Our method relies on a correlation filter embedding a fusion strategy and a motion trajectory correction scheme. The developed technique enhances the features' effective representation by extracting depth features and utilizes an improved Kalman filter to mitigate the tracking drifts in satellite videos through trajectory compensation and correction.

For the experiments, we choose eight satellite videos. Considering tracking moving vehicles and planes, our proposed algorithm affords a better tracking performance against current tracking algorithms, solving the problem of object occlusion. Compared with the SiamRPN tracking algorithm, the tracking accuracy of our method is slightly lower in some cases but still affords a good tracking performance.

The applications related to video satellites will become more and more widespread, and object tracking in satellite videos will gradually develop depending on the various needs. However, despite current methods affecting object tracking, additional work and innovation are acquired to improve accuracy, robustness, and object tracking performance. Although our method can solve the problem of low object tracking accuracy or tracking failure in satellite videos with complex backgrounds and object occlusion, there are some limitations. Firstly, suppose the object has been occluded before the adaptive Kalman filter converges. In that case, the prediction of the KCF tracker is not accurate when the object is occluded because the Kalman filter has not yet converged, which will result in low accuracy of the object tracking results or even tracking failure. Secondly, using hand-crafted features (HOG features) and depth features for object tracking is more computationally intensive and difficult to achieve real-time tracking on satellites. It would be very meaningful to design a robust feature with less computational complexity so that it is possible to track objects in real-time on satellites. Future work shall focus on object rotation and scale variation problems to improve object tracking accuracy further.

Author Contributions: Conceptualization, Y.L. (Yaosheng Liu) and Y.L. (Yurong Liao); methodology, Y.J. and C.L.; writing—original draft, Y.L. (Yaosheng Liu) and C.L.; writing—review and editing, Y.L. (Yaosheng Liu), Z.L. and X.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (NSFC) (No. 61805283).

Acknowledgments: The authors are grateful to Chang Guang Satellite Technology Co., Ltd. for providing the satellite videos used in our paper, without which our research would not have been possible. The authors also thank the anonymous reviewers and Academic Editors for providing valuable comments, which have been very beneficial for improving the manuscript. The authors would like to express their gratitude to EditSprings (https://www.editsprings.com/ (accessed on 20 November 2021)) for the expert linguistic services provided.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yilmaz, A.; Javed, O.; Shah, M. Object tracking: A survey. *ACM Comput. Surv.* **2006**, *38*, 1–45. [CrossRef]
2. Chen, X.; Xiang, S.; Liu, C.; Pan, C. Vehicle Detection in Satellite Images by Hybrid Deep Convolutional Neural Networks. *IEEE Geosci. Remote Sens. Lett.* **2014**, *11*, 1797–1801. [CrossRef]
3. Kopsiaftis, G.; Karantzalos, K. Vehicle detection and traffic density monitoring from very high resolution satellite video data. In Proceedings of the 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Milan, Italy, 26–31 July 2015; pp. 1881–1884.

4. Yang, T.; Wang, X.; Yao, B.; Li, J.; Zhang, Y.; He, Z.; Duan, W. Small Moving Vehicle Detection in a Satellite Video of an urban Area. *Sensors* **2016**, *16*, 1528. [CrossRef] [PubMed]
5. Shao, J.; Du, B.; Wu, C.; Zhang, L. Tracking Objects from Satellite Videos: A Velocity Feature Based Correlation Filter. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 7860–7871. [CrossRef]
6. Shao, J.; Du, B.; Wu, C.; Zhang, L. Can We Track Targets from Space? A Hybrid Kernel Correlation Filter Tracker for Satellite Video. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 8719–8731. [CrossRef]
7. Du, B.; Sun, Y.; Cai, S.; Wu, C.; Du, Q. Object Tracking in Satellite Videos by Fusing the Kernel Correlation Filter and the Three-Frame-Difference Algorithm. *IEEE Geosci. Remote Sens. Lett.* **2017**, *15*, 168–172. [CrossRef]
8. Guo, J.; Yang, D.; Chen, Z. Object Tracking on Satellite Videos: A Correlation Filter-Based Tracking Method with Trajectory Correlation by Kalman Filter. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 3538–3551. [CrossRef]
9. Xuan, S.; Li, S.; Han, M.; Wan, X.; Xia, G. Tracking in Satellite Videos by Improved Correlation Filters with Motion Estimations. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 1074–1086. [CrossRef]
10. Bolme, D.; Beveridge, J.; Draper, B.; Lui, Y. Visual object tracking using adaptive correlation filters. In Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, 13–18 June 2010.
11. Henriques, J.; Caseiro, R.; Martins, P.; Batista, J. Exploiting the Circulant Structure of Tracking-by-Detection with Kernels. In Proceedings of the 2012 IEEE Conference on European Conference on Computer Vision (ECCV), Florence, Italy, 7–13 October 2012; pp. 702–715.
12. Henriques, J.; Caseiro, R.; Martins, P.; Batista, J. High-speed tracking with kernelized correlation filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 583–596. [CrossRef] [PubMed]
13. Wang, Q.; Fang, J.; Yuan, Y. Multi-cue based tracking. *Neurocomputing* **2014**, *131*, 227–236. [CrossRef]
14. Yin, Z.; Collins, R. Object tracking and detection after occlusion via numerical hybrid local and global mode-seeking. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Anchorage, AK, USA, 23–28 June 2008.
15. Du, S.; Wang, S. An Overview of Correlation-Filter-Based Object Tracking. *IEEE Trans. Comput. Soc. Syst.* **2022**, *9*, 18–31. [CrossRef]
16. Zhang, S.; Lu, W.; Xing, W.; Zhang, L. Learning Scale-Adaptive Tight Correlation Filter for Object Tracking. *IEEE Trans. Cybern.* **2020**, *50*, 270–283. [CrossRef] [PubMed]
17. Danelljan, M.; Hager, G.; Khan, F.; Felsberg, M. Learning spatially regularized correlation filters for visual tracking. In Proceedings of the 2015 IEEE Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015.
18. Danelljan, M.; Robinson, A.; Khan, F.; Felsberg, M. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In Proceedings of the 2016 IEEE Conference on European Conference on Computer Vision (ECCV), Zurich, Switzerland, 11–14 October 2016; pp. 472–488.
19. Danelljan, M.; Hager, G.; Khan, F.; Felsberg, M. ECO: Efficient convolution operators for tracking. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
20. Zhang, J.; Ma, S.; Sclaroff, S. MEEM: Robust tracking via multiple experts using entropy minimization. In Proceedings of the 2014 IEEE Conference on European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; pp. 188–203.
21. Danelljan, M.; Hager, G.; Khan, F.; Felsberg, M. Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
22. Danelljan, M.; Hager, G.; Khan, F.; Felsberg, M. Accurate scale estimation for robust visual tracking. In Proceedings of the 2014 British Machine Vision Conference (BMVC), Nottingham, UK, 1–5 September 2014.
23. Li, Y.; Zhu, J. A scale adaptive kernel correlation filter tracker with feature integration. In Proceedings of the 2014 IEEE Conference on European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; pp. 254–265.
24. Danelljan, M.; Khan, F.; Felsberg, M.; De, J. Adaptive color attributes for real-time visual tracking. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014.
25. Danelljan, M.; Hager, G.; Khan, F.; Felsberg, M. Convolutional features for correlation filter based visual tracking. In Proceedings of the 2015 IEEE International Conference on Computer Vision Workshops (ICCVW), Santiago, Chile, 7–13 December 2015; pp. 621–629.
26. Ma, C.; Huang, J.; Yang, X.; Yang, M. Hierarchical convolutional features for visual tracking. In Proceedings of the 2015 IEEE Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015.
27. Kalal, Z.; Mikolajczyk, K.; Matas, J. Tracking-learning-detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 1409–1422. [CrossRef] [PubMed]
28. Danelljan, M.; Hager, G.; Khan, F.; Felsberg, M. Discriminative scale space tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1561–1575. [CrossRef] [PubMed]
29. Bertinetto, L.; Valmadre, J.; Henriques, J.; Vedaldi, A.; Torr, P. Fully-convolutional Siamese network for object tracking. In Proceedings of the 2016 IEEE Conference on European Conference on Computer Vision (ECCV), Zurich, Switzerland, 11–14 October 2016.
30. Li, B.; Yan, J.; Wu, W.; Zhu, Z.; Hu, X. High performance visual tracking with Siamese region proposal network. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.

31. Wu, Y.; Lim, J.; Yang, M. Online object tracking: A benchmark. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 23–28 June 2013.
32. Wu, Y.; Lim, J.; Yang, M. Object tracking benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1834–1848. [CrossRef] [PubMed]