Object Tracking in Satellite Videos by Improved Correlation Filters With Motion Estimations

Shiyu Xuan, Shengyang Li, Mingfei Han, Xue Wan, and Gui-Song Xia, *Senior Member, IEEE*

***Abstract*— As a new method of Earth observation, video satellite is capable of monitoring specific events on the Earth's surface continuously by providing high-temporal resolution remote sensing images. The video observations enable a variety of new satellite applications such as object tracking and road traffic monitoring. In this article, we address the problem of fast object tracking in satellite videos, by developing a novel tracking algorithm based on correlation filters embedded with motion estimations. Based on the kernelized correlation filter (KCF), the proposed algorithm provides the following improvements: 1) proposing a novel motion estimation (ME) algorithm by combining the Kalman filter and motion trajectory averaging and mitigating the boundary effects of KCF by using this ME algorithm and 2) solving the problem of tracking failure when a moving object is partially or completely occluded. The experimental results demonstrate that our algorithm can track the moving object in satellite videos with 95% accuracy.**

***Index Terms*— Correlation filter, motion estimation (ME), object tracking, satellite videos.**

I. Introduction

THE launch of video satellites has enabled us to observe and measure moving objects on the Earth's surface, which provides rich information for monitoring rapid-changing events, such as oil reserve detection, disaster monitoring, ocean monitoring, ecosystem disturbance monitoring, and traffic condition monitoring [1], [2]. For instance, the Jilin-1 satellite constellation launched by China can continuously obtain 10–30 images/s of the same area and has played an important role in the urban investigation of China [2]–[4].

Among the analysis of satellite videos, moving object detection and tracking is highly demanded, which targets to locate moving objects on the surface and compute their trajectories [3]–[6]. The moving objects in satellite videos mainly include motor vehicles, airplanes, and ships. In contrast with video surveillance on the ground, object tracking in satellite videos is more difficult because of the following facts.

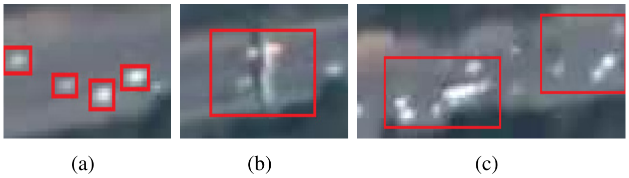


Fig. 1. Some moving objects in satellite videos. (a) Object size in the image is small about 10 × 10 pixels. (b) Object is partially occluded. (c) Small area with similar densely packed objects.

1) Due to the low spatial resolution, the size of the object is often small and the background is usually blurred and cluttered; therefore, many features that are suitable for tracking objects in natural scenarios quickly lose their efficiency for satellite videos.
2) Due to the bird's view, many moving objects in satellite videos are partially or completely occluded as shown in Fig. 1(b), which often leads to losses of objects.
3) As many similar objects are densely packed in a small area, the algorithm needs to have a very strong ability to distinguish targets.

See Fig. 1 for an instance, these difficulties have brought to light the need to study the tracking algorithm in order to resolve these problems.

The correlation filter method, named kernelized correlation filter (KCF) [7], has been reported promising results on tracking objects without rapid deformation, and therefore, it is among the best choices for tracking objects in satellite videos. However, when being applied to satellite videos, the KCF tracker still suffers from the following drawbacks: 1) when the object is not in the center of the searching area, the boundary effect often leads to decrease in the tracking accuracy and 2) object is often lost when it is completely occluded.

The moving objects in the satellite videos are vehicles, planes, or ships. The acceleration of these objects is not particularly large, which means that the motion state of these objects does not change in a short period of time. Therefore, the movement of objects in satellite video is usually regular. Inspired by this observation, this article presents a tracking algorithm based on the correlation filter with motion estimation (CFME) by integrating the movement characteristics of objects in the KCF tracker. More precisely, our work is distinguished from others with the following contributions.

1) We propose a ME algorithm combining the Kalman filtering and motion trajectory averaging (MTA).

Manuscript received March 18, 2019; revised July 22, 2019; accepted August 21, 2019. Date of publication October 23, 2019; date of current version January 21, 2020. This work was supported in part by the National Key Laboratory for Remote Sensing Information and Image Analysis Technology under Grant Y8180711WN. *(Corresponding author: Shengyang Li.)*

S. Xuan and M. Han are with the Key Laboratory of Space Utilization, Technology and Engineering Center for Space Utilization, Chinese Academy of Sciences (CAS), University of Chinese Academy of Sciences, Beijing 100094, China (e-mail: xuanshiyu17@csu.ac.cn; hanmingfei16@csu.ac.cn).

S. Li and X. Wan are with the Key Laboratory of Space Utilization, Technology and Engineering Center for Space Utilization, Chinese Academy of Sciences (CAS), Beijing 10094, China (e-mail: shyli@csu.ac.cn; wanxue@csu.ac.cn).

G.-S. Xia is with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China (e-mail: guisong.xia@whu.edu.cn).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2019.2943366

2) We propose an effective solution to mitigate the boundary effect in the KCF tracker. Compared with other methods, see [8]–[10] for mitigating the boundary effect, our method utilizes the motion feature of moving objects in satellite videos without sacrificing the computational efficiency.
3) Our algorithm can track objects in satellite videos with 95% accuracy at approximately 120 frames/s (FPS), which achieved the state-of-the-art performance. Moreover, it can well tackle the problem of tracking failure when the object is completely occluded.

II. Related Work

A. Moving Object Tracking

Algorithm of moving object tracking can be divided into two categories, generative methods [11]–[18] and discriminative methods [7]–[10], [19]–[35]. The generative method constructs the object template by extracting the object features and uses the search algorithm in the next frame to get the position with the highest similarity to the object template. The generative methods need to find an effective expression of the object and an efficient search algorithm. The absence of background information will lead to object drift when the object is similar to the background. Another method, the discriminative method, overcomes these shortcomings very well.

The biggest difference between discriminative methods and generative methods is that discriminative methods train a classifier to distinguish between the object and the background instead of just focusing on the object itself. The addition of more information, especially background information, makes the discriminative methods more robust. The key part of the discriminative method is the classifier, which needs to correctly classify the object and the background. In object tracking, the classifier is generally trained using the image of the first frame. After that, many algorithms use the new tracking results to expand the training set and uses the new training set to update the classifier online to make the classifier more robust.

One type of discriminative method uses the machine learning methods to train the classifier [19]–[25]. In recent years, deep learning has also been used in the field of object tracking [26], [27], [36]–[41]. The use of deep features avoids complex feature design and achieves good results, but deep learning tracking algorithm has difficulty in solving the problem of online updating because of the computational complexity. Tracking small size objects in satellite video using deep learning is a problem worth studying.

The correlation filter is another type of discriminative tracking methods [7]–[10], [26]–[33]. The correlation filter algorithm uses a cyclic shift to generate training samples and can be efficiently calculated in the Fourier domain, making the algorithm very efficient. Its high efficiency and high accuracy make it widely studied in the field of object tracking in recent years. The correlation filter called the minimum output sum of squared error (MOSSE) was applied to object tracking for the first time in [28]. The circulant structure kernel (CSK) tracker [32] used a cyclic shift to construct samples and added the kernel trick. On the basis of CSK, the KCF tracker [7] defined the connection method of multichannel features, and the filter can use different kernel functions. The scale adaptation of the correlation filter was added in [30] and [33] by using scale pyramids. The deep features were used in [26] and [38] to replace the handcraft features used by the general correlation filter method and achieved good results. The C-COT [31] used the interpolation to obtain a better fusion of different features and improved the accuracy but the efficiency was still low. The ECO [29] speeded up C-COT and improved accuracy by using dimensionality reduction and sample merging. The ECO is one of the best tracking algorithms at present.

The boundary effect is one of the important reasons that affect the performance of the correlation filtering algorithm. The samples generated by cyclic shift do not really contain background information. Therefore, the number of samples is not enough to train a robust classifier leading to the overfitting issue and the decrease in performance. When the moving object is deformed or occluded, the accuracy of the classifier will drop rapidly. In order to smooth the boundary of the samples, the samples need to be multiplied by the window function. When the object is not in the center of the search area, part of the information will be lost, which also increases the boundary effect.

The two commonly used methods for mitigating the boundary effect have been proposed by Danelljan *et al.* [8] and Galoogahi *et al.* [9]. A larger area for training was used, and spatial regular terms to reduce sample weights away from the target center were added in [8]. This method alleviates the boundary effect, but the filter coefficient can only be solved by iteration. It greatly increases the computational complexity, and the FPS is only about 5. In [9], the boundary effect was alleviated by adding real background information to the training sample, but the computational complexity is still high.

Different from the above two method, CFME provides another way to alleviate the boundary effect according to the object feature of satellite video, using the ME method to keep the object in the center of the search area. The increase in the computational complexity of our method is very low, which can retain the advantage of the high speed of the correlation filter.

B. Object Tracking in Satellite Videos

Most current algorithms for object tracking in satellite video are based on generative object tracking methods or moving object detection methods. Based on the assumption that the motion of a vehicle is linear in a short time, a method of template matching by using the Hu matrix after preprocessing with motion smoothing constraints was proposed in [3]. This method performed well under the simple road conditions but is not good under the complicated road conditions due to the object drift issue. A robust automatic detection and tracking method for video data of the UrtheCast Iris camera installed on the Zvezda module of the International Space Station (ISS) was proposed in [42]. This method uses background subtraction and motion recognition technology to detect and track motor vehicles and ships. This method requires high accuracy of the background subtraction algorithm and may fail in the presence of strong background jitter. Both spectral and spatial features were used to model ships and planes in [6] and in the object matching procedure, the Bhattacharyya distance, histogram intersection, and pixel count similarity were combined in a novel regional operator design. This method uses a worldview-2 multiangle time-series image to track the object and achieves good results in tracking plane objects; however, this method is not suitable for small vehicles.

Some algorithms use optical flow to tracking object in satellite video. In [4], the optical flow was used to track the vehicles and achieved vehicle speed estimation and traffic density monitoring; however, the main purpose in that article is to estimate the density of traffic and the tracking accuracy is not high. In [43], the optical flow was fused with the HSV color system and integral image, and the multiframe difference method was also used to get a better tracking results. This algorithm works very well when tracking the large size object. The calculation of the optical flow has a high requirement of video quality. Therefore, this algorithm is not effective when the video quality is not high.

Some algorithms use more robust discriminative methods and achieve very good performance. In [5], the correlation filter was used, and the multiframe difference method was used to assist the correlation filter. In [44], the optical flow was used as features to train the correlation filter. These two algorithms improve the correlation filter according to the object feature of satellite video and achieve very good results. However, these two algorithms do not solve the defects of the correlation filter itself like the boundary effect and will lose the object when the object is completely occluded. Different from these two algorithms, our methods use a simple method to alleviate the defects of the correlation filter itself.

As a new Earth observation technology, satellite video has been used for moving object detection in most studies. In the aspect of tracking, the methods perform well under many simple conditions, but current algorithms still need to be improved in terms of accuracy, robustness, and real-time performance.

III. Kernel Correlation Filter Tracker

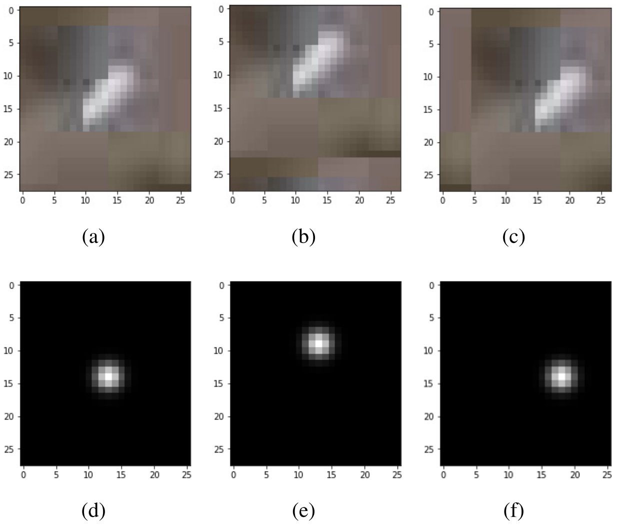
Assuming that the data are $x = [x_1, x_2, x_3, \ldots, x_n]$, a cyclic shift of data can be expressed as $P_x = [x_n, x_1, x_2, \ldots, x_{n-1}]$. All cyclic shifts of this data can be concatenated as data matrix $X = C(x)$ as shown in the following equation. This matrix is called a circulant matrix:

$$\begin{pmatrix} x_1 & x_2 & x_3 & \cdots & x_n \\ x_n & x_1 & x_2 & \cdots & x_{n-1} \\ x_{n-1} & x_n & x_1 & \cdots & x_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_2 & x_3 & x_4 & \cdots & x_1 \end{pmatrix}. \quad (1)$$

All circulant matrices have the following property [45]:

$$X = F^H \mathrm{diag}(\sqrt{n}Fx)F \quad (2)$$

where F is the discrete Fourier transform (DFT) matrix and is used for transforming the data to the Fourier domain. F^H is the Hermitian transpose of F. diag means the diagonal matrix. The solution of the linear regression can be simplified by using the properties of a circulant matrix.



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 2. Samples produced by cyclic shift and corresponding label functions. (a) Original image of object. (b) Image with a vertical upward cyclic shift of five pixels. (c) Image with a horizontal right cyclic shift of five pixels. (d)–(f) Label functions corresponding to the upper sample. It can be observed that because of the cyclic shift, a very sharp boundary appears in (b) and (c). In order to alleviate this situation, the image needs to be multiplied by the window function.

KCF uses the form of ridge regression to solve filter coefficients. Suppose the size of an image patch x is $M \times N$ pixels. The training samples are the all circular shifts of the image patch $x_{m,n}$, with $(m, n) \in \{0, 1, \ldots, M-1\} \times \{0, 1, \ldots, N-1\}$. The label function $y = \exp\{-((m-M/2)^2 + (n-N/2)^2/2\sigma^2)\}$ is a Gaussian function and (m, n) indicates the shifted positions along the horizontal and vertical directions, and the value at the target center is 1 and decays to 0 as the distance from the target center increasing. The samples and labels are shown in Fig. 2. The objective function can be written as

$$\min_{\omega} \sum_{i=1}^{M \times N} (f(x_i) - y_i)^2 + \lambda \|\omega\|^2 \quad (3)$$

where $f(x) = \omega^T x$ and $\lambda > 0$ is a regularization factor.

Then, ω can be calculated directly through

$$\omega = (X^T X + \lambda I)^{-1} X^T y. \quad (4)$$

Using the properties of the Fourier transform and substituting (2) into (4), we obtain

$$\hat{\omega} = \frac{\hat{x}^{\star} \circ \hat{y}}{\hat{x}^{\star} \circ \hat{x} + \lambda} \quad (5)$$

where $\hat{\omega}$, $\hat{x}$, and $\hat{y}$ are the DFT of ω, x, and y, respectively. $\hat{x}^{\star}$ is the complex conjugation of $\hat{x}$. The operator $\circ$ is the Hadamard product of matrix. λ is the regularization factor for ridge regression. By using the properties represented in (2), KCF does not need to generate cyclic shift samples by iteration. Using the Fourier transform of the original sample and (5), we can get the same result as what cyclic shift generates.

The performance of the tracker can be improved by using the kernel trick [7]. Suppose the kernel κ is $\kappa(x, x') = \langle \Phi(x), \Phi(x') \rangle$. $f(x)$ can be written as

$$f(x) = \omega^T \, \Phi(x) = \sum_{i=1}^{n} \alpha_i \kappa\left(x_i, x_i'\right). \tag{6}$$

For most of the kernel functions, the properties of (2) still hold. Then, $\hat{\alpha}$ can be solved by

$$\hat{\alpha} = \frac{\hat{y}}{\hat{k}^{xx'} + \lambda}. \tag{7}$$

If a Gaussian kernel is used, $k^{xx'}$ can be written as (8), where c represents the number of channels of the feature

$$k^{xx'} = \exp\left\{ -\frac{1}{\sigma^2} \left(\|x\|^2 + \|x'\|^2 - 2F^{-1} \left(\sum_c \hat{x}_c^{\star} \circ \hat{x}_c' \right) \right) \right\}. \tag{8}$$

In the tracking process, the algorithm crops an image patch with 2.5 times the object size in the object center to obtain features x. $\hat{\alpha}$ can be calculated in (7). In the next frame, the features z are extracted from the image patch, which is cropped at the center at the object center of the previous frame with the size unchanged. The algorithm then calculates the correlation response patch of this frame using

$$f(z) = F^{-1}(\hat{k}^{lz} \circ \hat{\alpha}) \tag{9}$$

where l is the learned target template. The object position of the current frame can be obtained by calculating the offset of the maximum value position from the center of the response patch.

In order to ensure that the filter can adapt to the changes of the object, we need to update the filter by using (7) for training the filter α_{new} with the samples x_{new}, which are sampled from the new position. The new filter coefficients are calculated as

$$\begin{cases} \hat{\alpha} = (1 - \eta)\, \hat{\alpha}_{t-1} + \eta\, \hat{\alpha}_{\text{new}} \\ l_t = (1 - \eta)\, l_{t-1} + \eta\, x_{\text{new}} \end{cases} \tag{10}$$

where η is the learning rate, α_{new} is calculated using new samples, and l_t is the new learned target template.

KCF has low computational complexity and can use multichannel features, e.g., histogram of oriented gradients (HOG) and color naming. It is robust to illumination changes and has high tracking accuracy for objects without rapid deformation. It is very suitable for moving object tracking in satellite video.

IV. Proposed Method

In order to achieve high accuracy moving object tracking in satellite videos, we propose a correlation filter with the ME algorithm (CFME) (see Fig. 3). In this section, we first introduce the ME algorithm, then introduce the use of ME to mitigate the boundary effect of KCF and the problem of tracking failure because of occlusion.

Algorithm 1 ME Algorithm $w, P_{new\_estimate} \leftarrow ME(frames, n, P_{old})$

```
Input:
$frames$: video stream;
$n$: number of processed frames;
$P_{old}$: the object position of previous frame;

Output:
$w$ whether the motion estimation is work;
$P_{new\_estimate}$ estimated new position;

if $n == 1$ then
  /*do some initialization*/
  Initialize the Kalman filter $Kalman$;
  $kn \leftarrow 0$;
  flag $\leftarrow$ FALSE;
  Set $ta_n$(the number of frames used for motion trajectory
  averaging);
  return $w \leftarrow$ FALSE, $P_{new\_estimate} \leftarrow$ None
else if $i < ta_n$ then
  return $w \leftarrow$ FALSE, $P_{new\_estimate} \leftarrow$ None
else
  if flag == FALSE then
    Calculate $P_{estimate}$ by using Equation (20)-(22) and
    $P_{old}$;
    $P_{new} \leftarrow CFME(frames)$;
    Use $Kalman$ and $P_{old}$ get $P_{k\_estimate}$;
    if distance($P_{k\_estimate}$, $P_{new}$) < 4 then
      $kn \leftarrow kn + 1$;
      if $kn == 4$ then
        flag $\leftarrow$ TRUE;
      end if
    else
      $kn \leftarrow 0$;
    end if
    $P_{new\_estimate} \leftarrow P_{estimate}$;
    return $P_{new\_estimate}$, $w \leftarrow$ FALSE
  else
    Use $Kalman$ and $P_{old}$ get $P_{k\_estimate}$;
    $P_{new\_estimate} \leftarrow P_{k\_estimate}$;
    return $P_{new\_estimate}$, $w \leftarrow$ FALSE
  end if
end if
```

A. Motion Estimation

1) Kalman Filter: In this article, we use the Kalman filter (KF) [46] to estimate the position and the velocity of moving objects. The state equation and observation equation of the system can be written as

$$X_k = \phi_{x,k-1} X_{k-1} + W_{k-1} \tag{11}$$

$$Y_k = H_k X_k + V_k \tag{12}$$

where X_k and X_{k-1} are the state vectors of the system at time k and $k - 1$, respectively. $\phi_{x,k-1}$ is the state transition matrix of the system, and H_k is the observation matrix of the system. W and V are noise matrices following the Gaussian

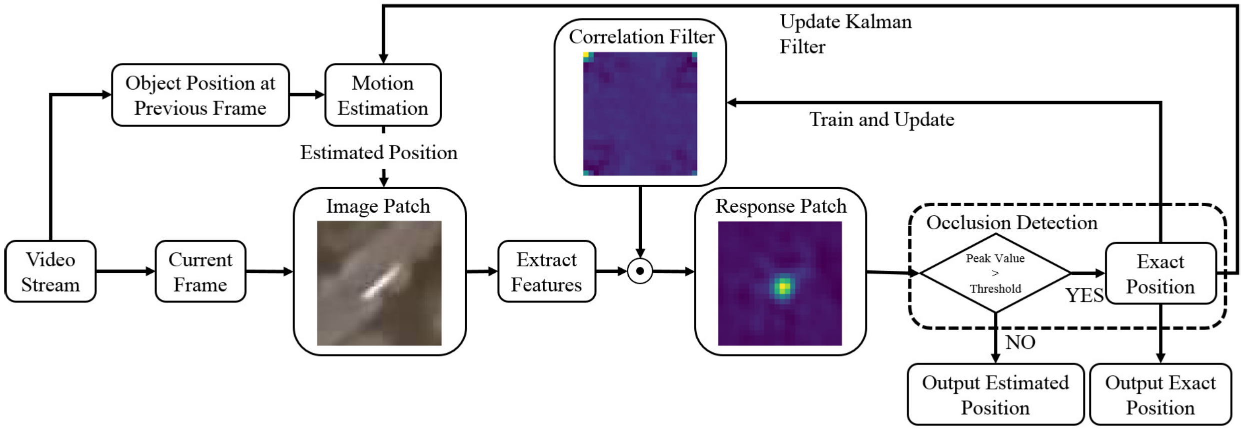


Fig. 3. Pipeline of the proposed CFME. First, we use the object position at a previous frame to estimate the position. Then, we crop image patch and extract features at the estimated position. Third, we use the method mentioned in Section III to obtain the response. After that, we need to judge whether the object is occluded. If the object is occluded the position predicted by correlation filter is inaccurate and the estimated position will be used as the output; otherwise, the position predicted by the correlation filter will be used as the output.

distribution with covariance matrices Q and R. In this system, we select the state vector as $X_k = (xs_k, ys_k, \Delta x_k, \Delta y_k)^T$ where xs_k and ys_k are the horizontal and vertical positions of the object at time k, respectively, and Δx_k and Δy_k are the horizontal and vertical speeds of the object at time k, respectively.

Since the time between every two frames is short, it can be assumed that the moving object such as the vehicle is moving in a uniform linear motion; therefore, the state transition matrix can be written as follows:

$$\phi_{k,k-1} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{13}$$

The observation vector is $Y_k = (xw_k, yw_k)^T$, which represents the object position observed at time k. H_k can be expressed as

$$H_k = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}. \tag{14}$$

We then use the KF for ME as follows:

$$\hat{X}_{k+1,k} = \phi_{k+1,k}\hat{X}_k \tag{15}$$

$$\hat{X}_{k+1} = \hat{X}_{k+1,k} + K_{k+1}(Y_{k+1} - H_{k+1}\hat{X}_{k+1,k}) \tag{16}$$

$$K_{k+1} = \hat{P}_{k+1,k}H_{k+1}^T\left(H_{k+1}P_{k+1,k}H_{k+1}^T + R_k\right)^{-1} \tag{17}$$

$$P_{k+1,k} = \phi_{k+1,k}P_k\phi_{k+1,k}^T + Q_k \tag{18}$$

$$P_{k+1} = (I - K_{k+1}H_{k+1})P_{k+1,k} \tag{19}$$

where $\hat{X}_{k+1}$ is the optimal state estimate, K is the KF gain matrix, and Q and R are the covariance matrices of noise, which can be adjusted according to the actual situation. P_0 is generally initialized with the random data that are not 0. I is the identity matrix.

The computation of the KF contains only matrix multiplication for ten times, matrix additional for five times, and an inverse of a 2 × 2 matrix. Compared with the computational complexity of KCF, the increased computational complexity is very little since the size of the biggest matrix in (15)–(19) is 4 × 4.

The KF needs some amount of data to converge. Experiments show that the KF can converge after 30–50 frames for ME of moving the object in satellite video.

2) Motion Trajectory Averaging: The KF has high accuracy in estimating object motion state, but KF is complex and the filter cannot converge until some frames are used to update the filter.

In order to estimate the object motion before the KF converges, we propose a method called MTA.

Typical moving objects in satellite videos are motor vehicles, planes, and ships. We can assume that in a short period of time, the object is moving in a uniform straight line, even if the object is in a state of turning, emergency stop or acceleration. Based on this assumption, the speed of the object in the current frame can be estimated by the average displacement of the previous frames. The position of the object at the current frame can be estimated using the speed and the position of the object in the previous frame. Thus, the MTA can be described in the following equations:

$$\Delta x_{t-1} = \frac{1}{n}\sum_{i=1}^{n}(x_{t-i} - x_{t-i-1}) \tag{20}$$

$$\Delta y_{t-1} = \frac{1}{n}\sum_{i=1}^{n}(y_{t-i} - y_{t-i-1}) \tag{21}$$

$$P_t = A\,S_{t-1} \tag{22}$$

where $S_{t-1} = (x_{t-1}, y_{t-1}, \Delta x_{t-1}, \Delta y_{t-1})^T$ is the state vector of the object at time $t-1$. $P_t = (x_t, y_t)^T$ is the position vector of the object at time t and A is a transfer matrix and can be written in the form as

$$A = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}. \tag{23}$$

In (20) and (21), n is the number of frames used for MTA and is determined by considering the FPS of the satellite video. If n is too small, the MTA will be too sensitive to changes in the object motion state. If n is too large, the assumption mentioned above is not true. Therefore, the value needs to be carefully selected.

3) Motion Estimation Algorithm With MTA and KF: The KF can converge only after a certain number of frames. Before the KF converges, we use the result of MTA as the output of ME. After the KF converges, we use the result of the KF as the output of ME.

The results of the ME will be used as the center of the search area of the KCF to determine the exact position of the object. The algorithm will be described in detail in Section IV-B. The exact position of the object obtained by the KCF will be used as the observation vector to update the KF. In order to determine whether the KF is convergent, the exact position of the object obtained by KCF will be compared with the estimated position of the KF. Considering that the KF is random when it does not converge, the KF is considered to be convergent only if the Euclidean distance between the estimated position of the KF and the exact position of the object obtained by KCF is less than four pixels in four consecutive frames. The setting of the distance has little effect on the experimental results as long as the distance is not set too large. The pseudocode of the ME algorithm can be seen in Algorithm 1.

B. Correlation Filter With Motion Estimation

1) Using Motion Estimation to Mitigate Boundary Effect: As described in Section II-A, KCF has a boundary effect. The method of [8]–[10] will greatly increase the computational complexity of the algorithm. We use ME to mitigate the boundary effect by placing the object being tracked in the center of the search area. According to the description of Section IV-A, the ME algorithm does not work with limited frames at the beginning of a video. We use the original KCF algorithm and use the center position of the object at the previous frame as the center of the search area of the current frame until the ME starts working. Then, we crop the image patch 2.5 times the object size around the center that was defined by the ME algorithm to extract features and then use (9) to calculate the exact position of the object. Using this method, the object being tracked can be kept at the center of the search area. The results of the experiment show that the accuracy of KCF has been greatly improved through this method.

2) Tracking Method When the Object Is Occluded: Compared to ordinary object tracking tasks, the situation where the object is completely occluded is very common in satellite videos. As shown in Fig. 4, when the vehicle passes the bottom of the overpass, the vehicle will be completely occluded and disappear from the image. Most trackers lose the object when it is occluded and cannot relocate the object when it appears again.

In order to track the object correctly when the object is occluded, the following three subproblems need to be solved.

1) *Occlusion Detection:* The algorithm needs to detect the occurrence of occlusion of the object.
2) *Occlusion Processing:* When the object is complete or large-area occluded, processing is required to ensure that the tracking algorithm does not lose the object.
3) *End of Occlusion Detection:* The algorithm needs to detect the end of occlusion.

Algorithm 2 CFME Tracker $P_{new} \leftarrow CFME(frames)$

```
Input:
$frames$: video stream;

Output:
$P_{new}$: the new position of object;

Set the occlusion threshold $T$;
for $i = 1; i <= len(frames); i++$ do
  if $i == 1$ then
    /*do some initialization in first frame*/
    /*select the object to track*/
    $P_{old} \leftarrow$ the position of the tracked object
    Initialize the KCF tracker $K$;
  else
    $w, P_{estimate} \leftarrow ME(frames, i, P_{old})$;
    if $w ==$ FALSE then
      Crop image patch from $frames[i]$ to get feature(size is 2.5 times object size and center is $P_{old}$);
      Use feature and $K$ calculate the position $P_{new}$;
      return $P_{new}$
    else
      Crop image patch from $frames[i]$ to get feature(size is 2.5 times object size and center is $P_{estimate}$);
      Use feature and $K$ calculate the position $P$ and the max value $p_v$ of the response map;
      if $p_v > T$ then
        /*no occlusion*/
        Update the Kalman filter in $ME$ and the KCF tracker $K$;
        return $P_{new} \leftarrow P$
      else
        /*the object is occluded*/
        return $P_{new} \leftarrow P_{estimate}$
      end if
    end if
  end if
end for
```

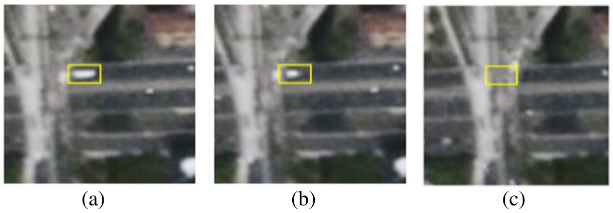

(a) (b) (c)

Fig. 4. Object is surrounded by a rectangle. (a) Target is not occluded. (b) Object is partly occluded. (c) Object is completely occluded and disappears from the image.

To solve the above problems, the following steps are taken in our algorithm.

1) As described in Section III, the higher the peak value of the response patch calculated by the KCF tracker, the more confidence the tracking result has. In general, the rapid illumination variation, the rapid deformation of object, the motion blur of the object, and the partial or

complete occlusion of the object can lead to a low peak value. In satellite video, the rapid illumination variation, the motion blur of object, and the rapid deformation of object are not obvious; Therefore, we can think that the low peak value is related to the partial or complete occlusion of object in most cases, and the peak value of the response patch can be used to determine whether the object is occluded. The object is occluded if the peak value is less than a threshold. We will describe how to choose a threshold in Section V-D.

2) When the target is occluded, the position calculated by the KCF tracker will be inaccurate and the position obtained by the KCF tracker needs to be discarded, and the position estimated by the ME is used as the position of the object. Compared with the KCF tracker, the accuracy of the position obtained by ME is limited; therefore, this position cannot be used to update the KF. At the same time, the KCF tracker also stops updating to prevent the filter from learning the occluded features.
3) When the peak value of the response patch obtained by the KCF tracker is greater than the threshold again, the object is considered to reappear and occlusion of the object has ended.

By using this method, when the object is occluded, the tracker can estimate the approximate position of the object. When the object appears again, the algorithm can quickly relocate the object to obtain the exact position of the object.

The pseudocode of the correlation filter with the ME algorithm is shown in Algorithm 2.

V. Experiment and Analysis

A. Data Sets and Compared Algorithms

The experimental data come from the Jilin-1 satellite constellation developed by China Changchun Satellite Technology Co., Ltd. We used 11 videos for experimentation. The spatial resolution of data is approximately 1 m, and the frame rate is 10. Among the videos, there are two airports in Frankfurt, Germany and Guizhou, China, and the moving objects are plane. The number of objects is two, and the objects size is about 50 × 40 pixels. The rest of the video observes traffic conditions in Dubai, Hong Kong, and Boston. The moving objects are motor vehicles and the number of objects is 11, the size of objects is large as 23 × 8 pixels, with the smallest object being 8 × 8 pixels.

We choose to use KCF [7], ECO [29], TLD [23], MIL [20], BOOSTING [21], and MEDIANFLOW [47] to compare with our CFME algorithm. KCF is the baseline of our algorithm. MIL and BOOSTING are the classic discriminative tracking algorithms. Both TLD and MEDIANFLOW use the optical flow method, and TLD uses redetection method to prevent the loss of object. ECO is one of the best correlation filter algorithms in object tracking.

B. Details on the Setting of Parameters

The CFME algorithm is implemented in Python. The TLD, MIL, BOOSTING, and MEDIANFLOW are implemented by calling opencv API. The source code of KCF is from http://www.robots.ox.ac.uk/ joao/circulant/index.html, and the source code of ECO is from http://www.cvl.isy.liu.se/research/objrec/visualtracking/ecotrack/index.html. All algorithms are performed on the computer with a 2.4-GHz intel Xeon E5 2620 v3 CPU. Both KCF and CFME use HOG features. The cell size of the HOG is 4×4. The regularization factor λ is set to 10^{-4}. The learning rate η is set to 0.012. The label function y is a 2-D Gaussian function, whose bandwidth is $\sqrt{wh}/16$. The variables w and h are the width and height of the tracked object, respectively. The size of the search area is 2.5 times the tracked object size. Considering the assumption that the object is moving in a uniform linear motion for a short period of time, n in (20) and (21) is set to 5 for a video whose frame rate is 10 Hz and is set to 10 if the frame rate of the video is 24 Hz.

C. Evaluation Metrics

There are two commonly used evaluation metrics center location error (CLE) and overlap score to evaluate the performance of tracker at each frame. CLE is the Euclidean distance between the groundtruth center of the tracked object and the predicted position. Given the predicted bounding box r_t and the groundtruth bounding box r_a, the overlap score is defined as

$$S = \frac{|r_t \bigcap r_a|}{|r_t \bigcup r_a|} \tag{24}$$

where $\bigcap$ and $\bigcup$ represent the intersection and union of two regions and denote the number of pixels in the region [48], [49].

Based on CLE and overlap score, in order to evaluate the performance of tracker on the whole video, precision plot, success plot, precision score, success score, and area under curve (AUC) are used [48], [49]. The precision plot shows the percentage of frames whose CLE is within the given threshold distance. Considering that the size of the moving object in satellite video is small, the tracking for a given frame is regarded as successful if CLE is within 5 pixels (20 pixels in [48] and [49]). The precision score for each tracker is the percentage of successful tracking frames. The success plot shows the percentage of frames whose overlap score is larger than a given threshold. The tracking is regarded as successful for a given frame if the overlap score is larger than the threshold 0.5. The success score is the percentage of successful tracking frames. We use the AUC of success plot to rank the trackers.

To evaluate the speed of the tracker, the FPS is also used. It is the number of frames that the tracker can process per second.

D. Setting of the Threshold to Detect Occlusion

In order to correctly select the threshold for occlusion detection in Algorithm 2, we used the peak value of the response patch of all video sequences to plot their distribution in Fig. 5. The distribution of the peak value is the bimodal. As described in Section IV-B2, the low peak value is related to the partial or complete occlusion of object in most cases.

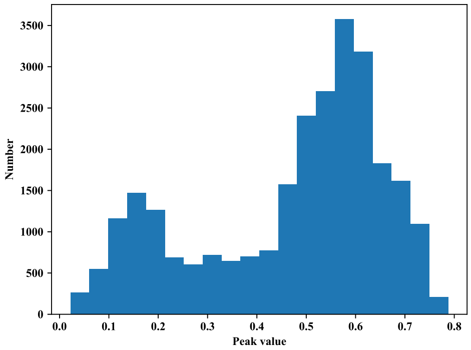


Fig. 5. Distribution of the peak value of the response function. We can see the obvious bimodal distribution. Therefore, occlusion and nonocclusion can be distinguished by threshold.

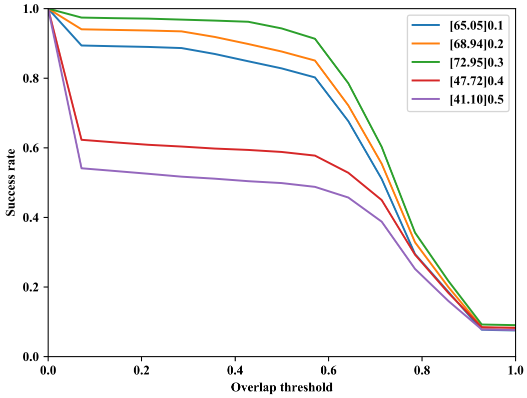


Fig. 6. Success plot on every threshold and the legend of the success plot is the AUC of each threshold.

We can only use the peak value to determine whether the object is occluded or not.

We used a grid search to select the best performing threshold from $[0.1, 0.2, 0.3, 0.4, 0.5]$. The success plot on every threshold is shown in Fig. 6. The AUC is highest when the threshold is 0.3, therefore we choose 0.3 as the threshold.

The visualization of the tracking process in Fig. 7 explains the relationship between the object state and the peak value. The threshold 0.3 can distinguish whether the object is occluded very well. The peak value is less than the threshold only when the object is seriously occluded. In the videos that the object is not occluded, the peak value is not less than the threshold. In a few extreme cases, the peak value below the threshold is not caused by the object being seriously occluded but caused by rapid illumination variation or the motion blur of the object. The duration of such extreme cases is very short and will bring negligible error to the final tracking results. Therefore, our improvement for occluded objects has little effect on the objects without occlusion.

E. Experimental Analysis on Moving Vehicles Tracking

The results of moving vehicles tracking are shown in Table I. The accuracy of our CFME algorithm greatly exceeds the other algorithms. Compared with KCF, the CFME improved the tracking accuracy and the AUC increased by approximately 14%, the success score increased by approximately 20% and the precision score increased by approximately 17%. The TLD algorithm and the MEDIANFLOW algorithm fail in tracking vehicles. The possible reason is that these two algorithms need to extract feature points to calculate the optical flow, but the feature points of the low-resolution vehicle are not obvious.

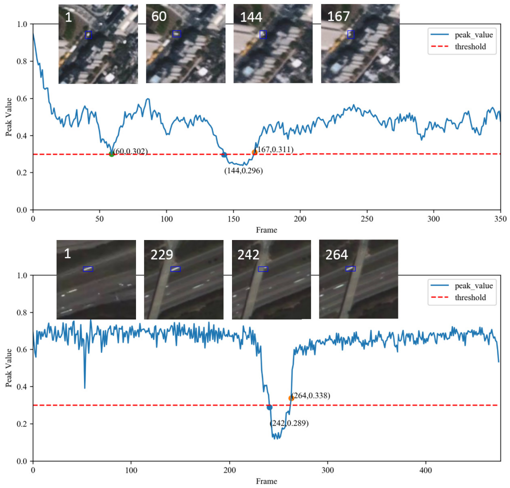


Fig. 7. Visualization of the tracking. As the peak value decreases, the more the object is occluded. When the peak value is very low (lower than 0.3), the object is seriously occluded almost disappears from the image.

In order to evaluate the performance of our algorithm when the object is occluded, we divided the satellite videos into two parts: the object is occluded and the object is unoccluded. The number of objects partially or completely occluded is four and the number of objects unoccluded is seven. We experiment with these two parts separately.

The results are shown in Fig. 8, Tables II and III. In the case of no occlusion, the accuracy of ECO is not as good as that of KCF, which may be caused by the scale adaptation of ECO. Compared with KCF, the CFME improves AUC by 8.5%, which proves the effectiveness of ME. In the case of occlusion, the accuracy of ECO exceeds KCF. When the target is partially occluded ECO can keep track of the object and show strong robustness when compared with KCF. However, when the object is completely occluded, all the algorithms except CFME lose the object. The performance of CFME is better than the other trackers for both occluded and unoccluded objects.

In terms of efficiency, although the CFME increases computational complexity because of ME, CFME is only slightly slower than KCF for about 7% and still faster than the other trackers.

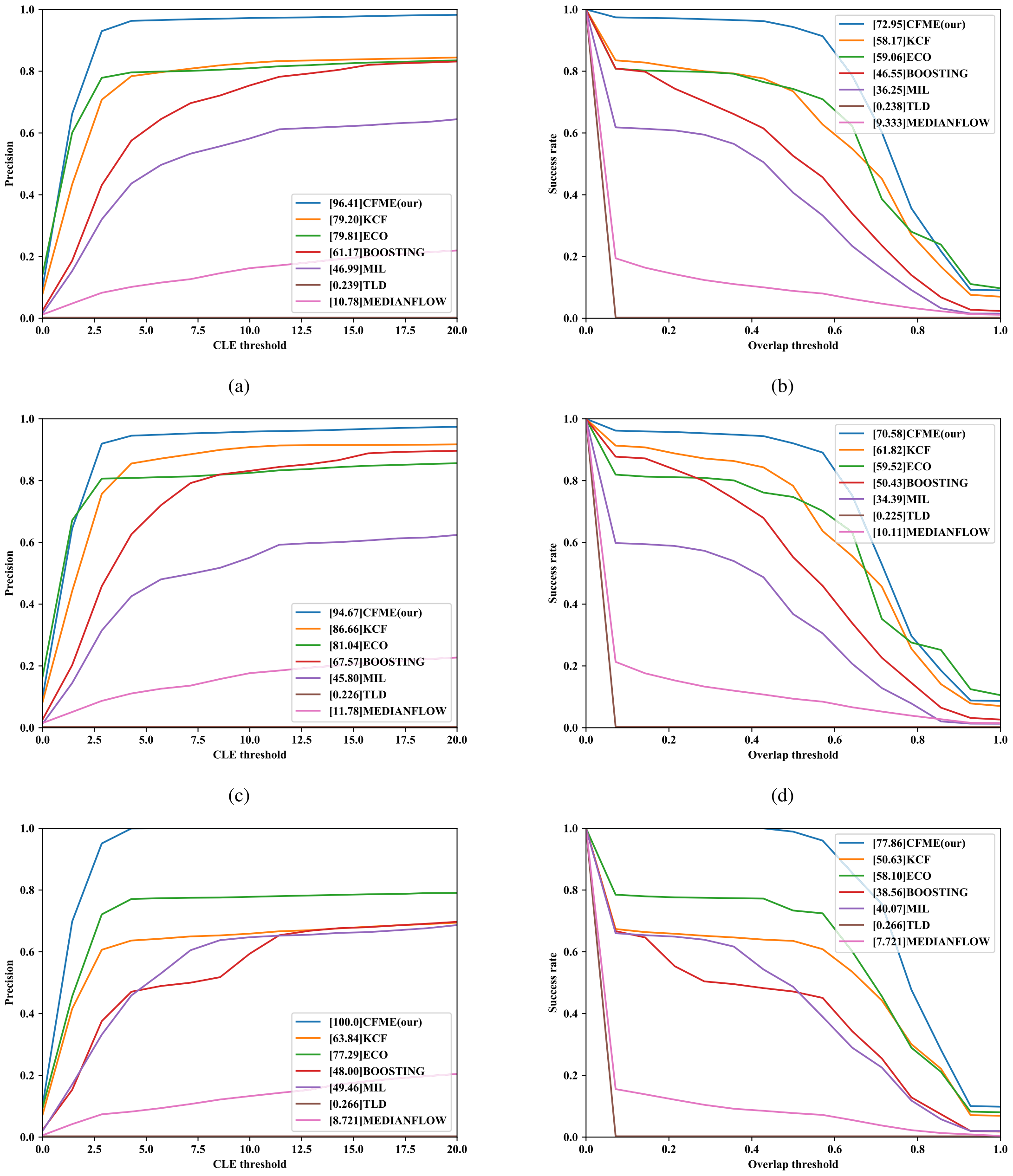
F. Experimental Analysis on Moving Plane Tracking

To test the adaptability of the algorithm to multiple types of objects, we use the plane as the tracking object. Because a plane at the airport will not be completely occluded, the occlusion detection of CFME is not necessary. Therefore, the threshold of occlusion detection is set to 0. The results are shown in Table IV and Fig. 9. It is shown that for the planes whose texture features and shape are clear, ECO performs best with AUC 76%. MEDIANFLOW is the fastest tracker and has an FPS of 155. The success score of MIL is 100%, but the precision score is particularly low and only 17.2%. This shows that although MIL can correctly track the object, the deviation

TABLE I

RESULTS OF VEHICLE TRACKING. OUR TRACKER OUTPERFORMS THE OTHER TRACKERS IN ALL AUC, SUCCESS SCORE, AND PRECISION SCORE. OUR TRACKER IS ONLY SLIGHTLY SLOWER THAN THE KCF TRACKER AND FASTER THAN THE OTHER TRACKERS. OUR TRACKER IS VERY GOOD AT TRACKING SMALL OBJECT IN SATELLITE VIDEO

	CFME (our)	KCF [7]	ECO [29]	BOOSTING [23]	MIL [20]	TLD [21]	MEDIANFLOW [47]
AUC	**72.9**	58.1	59.1	46.6	36.3	0.24	9.33
Success score	**94.3**	73.6	74.2	52.7	40.7	0.24	8.87
Precision score	**96.4**	79.2	79.8	61.2	47.0	0.24	10.8
FPS	123	**132**	58	61	7	2	87



(a) (b)

(c) (d)

(e) (f)

Fig. 8. Experimental results of moving vehicle tracking. (a), (c), and (e) Precision plots over all the sequences, object unoccluded sequences, and object occluded sequences, respectively. The legend of the precision plot is the precision score for each tracker. (b), (d), and (f) Success plots over all the sequences, object unoccluded sequences, and object occluded sequences, respectively. The legend of the success plot is the AUC for each tracker.

TABLE II

RESULTS OF VEHICLE TRACKING ON OCCLUDED DATA. OUR TRACKER OUTPERFORMS THE OTHER TRACKERS IN ALL AUC, SUCCESS SCORE, AND PRECISION SCORE. THE AUC OF OUR ALGORITHM IS ABOUT 20% HIGHER THAN THAT OF THE SECOND ONE. THE SUCCESS SCORE AND THE PRECISION SCORE PROVE THAT OUR TRACKER WILL NOT LOSE THE OBJECT EVEN IF THE OBJECT IS COMPLETELY OCCLUDED

	CFME (our)	KCF [7]	ECO [29]	BOOSTING [23]	MIL [20]	TLD [21]	MEDIANFLOW [47]
AUC	**77.9**	50.6	58.1	38.6	40.1	0.27	7.72
Success score	**98.9**	63.4	73.4	47.1	48.6	0.27	7.79
Precision score	**100**	63.8	77.3	48.0	49.5	0.27	8.72
FPS	121	**129**	56	58	6	1	84

TABLE III

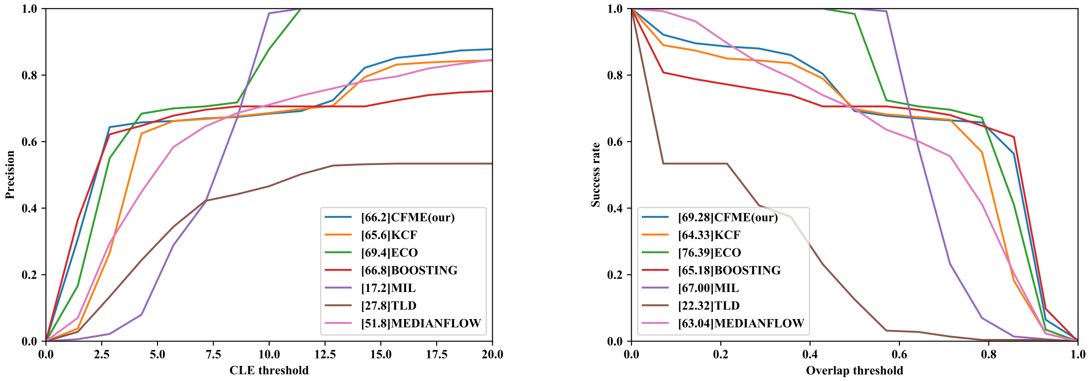
RESULTS OF VEHICLE TRACKING ON DATA WITHOUT OCCLUSION. OUR TRACKER OUTPERFORMS THE OTHER TRACKERS IN ALL AUC, SUCCESS SCORE, AND PRECISION SCORE. OUR TRACKER IS ALSO GOOD AT TRACKING THE OBJECT WITHOUT OCCLUSION

	CFME (our)	KCF [7]	ECO [29]	BOOSTING [23]	MIL [20]	TLD [21]	MEDIANFLOW [47]
AUC	**70.6**	61.8	59.5	50.4	34.4	0.23	10.1
Success score	**92.1**	78.3	74.7	55.4	36.8	0.23	9.40
Precision score	**94.7**	86.7	81.0	67.6	45.8	0.23	11.8
FPS	126	**139**	61	64	8	2	89

TABLE IV

RESULTS OF PLANE TRACKING. THE ECO TRACKER IS THE BEST IN PLANE TRACKING. THE AUC OF OUR TRACKER RANKS SECOND AND THE PRECISION SCORE RANKS THIRD. ALTHOUGH OUR TRACKER IS NOT THE BEST IN PLANE TRACKING, IT IS STILL COMPETITIVE IN TERMS OF ACCURACY AND SPEED

	CFME (our)	KCF [7]	ECO [29]	BOOSTING [23]	MIL [20]	TLD [21]	MEDIANFLOW [47]
AUC	69.3	64.3	**76.4**	65.2	67.0	22.3	63.1
Success score	69.2	69.8	98.2	70.6	**100**	12.6	69.8
Precision score	66.2	65.6	**69.4**	66.8	17.2	27.8	51.8
FPS	102	106	49	49	9	15	**155**



(a)

(b)

Fig. 9. Experimental results of moving plane tracking. (a) Precision plots. The legend of the precision plot is the precision score for each tracker. (b) Success plots. The legend of the success plot is the AUC for each tracker.

of the predicted position is large. The AUC of CFME is 69.3% and ranked second among all the trackers. The precision score is 66.2% ranked third. Although the success score is 69.2% ranked fifth, it is very close to the other trackers. Compared with KCF, the performance of CFME is better, which proves that our improvement of KCF is also effective in tracking large moving objects in satellite videos.

G. Qualitative Evaluation

Here, we provide a qualitative comparison of our approach with other trackers in Fig. 10. In the Boston and Atlanta, the object is completely occluded and disappears in the image. Our tracker is the only tracker that does not lose the object. When the object is completely occluded, the ME algorithm of our tracker can estimate the position of the object. When the object reappears, our tracker can relocate the object immediately. Other trackers have no ability to estimate the position of object when the object disappears. When the object is occluded for a long time, the object will leave the searching area of the trackers. Therefore, the trackers lose the object. In the Frankfurt, the object is not occluded in the whole video. Our tracker locates the object more accurately compared with the KCF because our tracker alleviates the boundary effect of KCF. In the Guizhou plane, the object is plane and

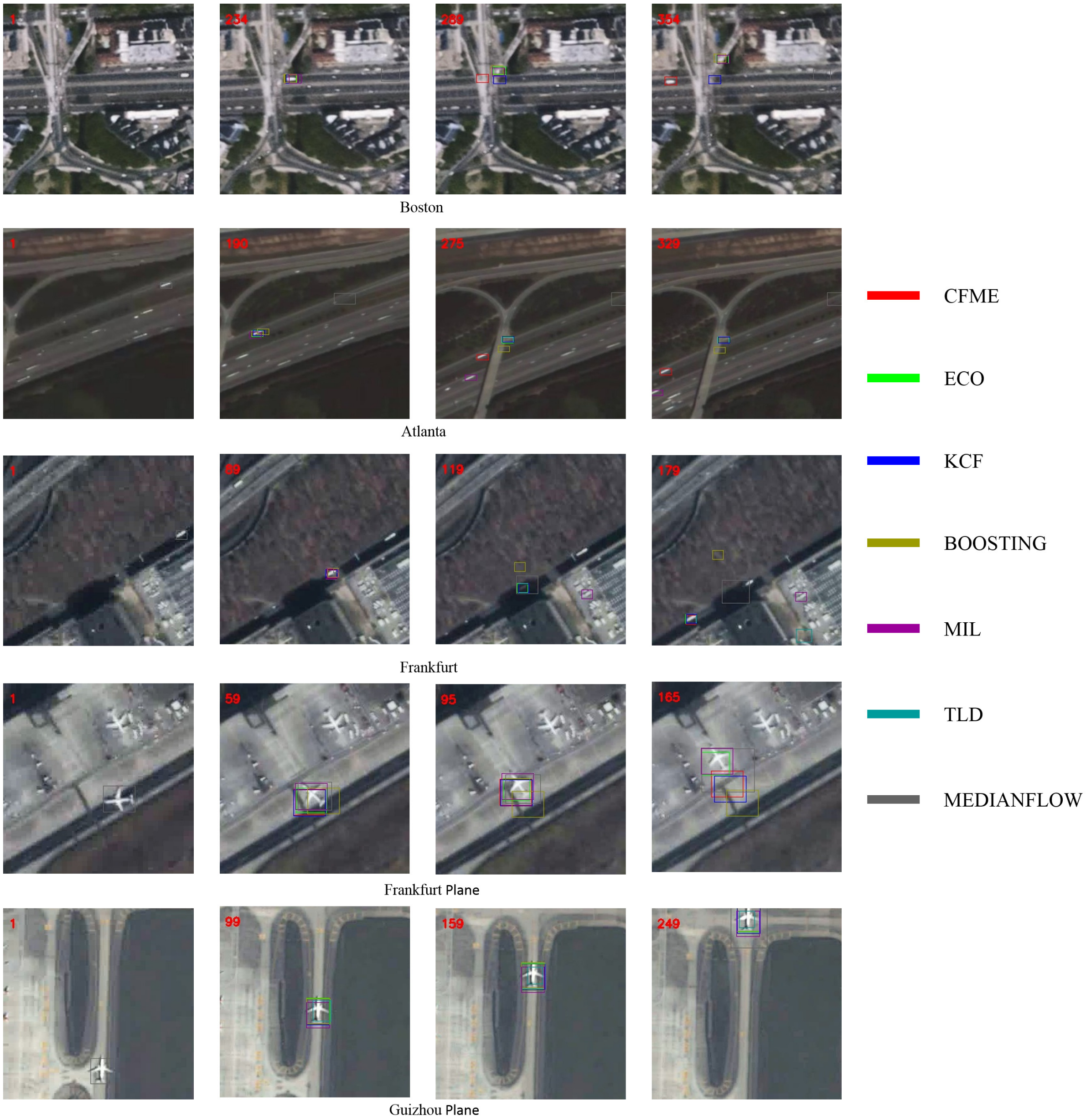


Fig. 10. Visualization of the tracking results. The number in the top-left corner of each image is the number of current frames in the video.

its size is larger than the vehicles. Therefore, this object is easier to track. The MEDIANFLOW, which does not work in tracking vehicle, can track plane because the corners of the plane are prominent. However, the ability of MEDIANFLOW to locate the object is poor. The performance of the CFME, KCF, and ECO is almost the same because the boundary effect of such a large object is not strong. In the Frankfurt plane, the object rotates in a wide range. This is the only video that our tracker loses the object. The features used by CFME and KCF are HOG. Our tracker and KCF lose the object since the HOG has no rotation invariance. Our tracker alleviates the boundary effect of the KCF. Therefore, our tracker is more robust than the KCF and lost the object later than the KCF. Besides HOG, the ECO also uses color names as its features. It does not lose the plane but the performance is also poor. Overall, the visual evaluation indicates that the effectiveness of our method to alleviate the boundary effect of the KCF and the ability of our tracker to track the completely occluded object.

VI. Conclusion

This article proposes an effective tracker called CFME based on the framework of the correlation filter. The CFME algorithm improves over KCF by ME through combining the MTA and the Kalman filtering. The CFME also mitigates the boundary effect and the problem of tracking occluded object under the premise of ensuring computational efficiency.

We conducted experiments on 11 satellite videos. For the moving vehicles tracking, CFME has a better performance in terms of speed and accuracy compared with the other trackers. CFME is the only tracker capable of tracking completely occluded objects. For tracking moving planes, the CFME has slightly lower accuracy than ECO but still has good performance in speed.

Overall, the CFME algorithm is very effective for tracking moving objects in satellite videos, especially small objects.

References

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

[2] T. Yang *et al.*, "Small moving vehicle detection in a satellite video of an urban area," *Sensors*, vol. 16, no. 9, p. 1528, 2016.

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

[4] Y. Luo, Y. Liang, and Y. Wang, "Traffic flow parameter estimation from satellite video data based on optical flow," *Comput. Eng. Appl.*, vol. 54, no. 10, pp. 204–207 and 255, 2018.

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

[6] L. F. Meng and J. P. Kerekes, "Object tracking using high resolution satellite imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 1, pp. 146–152, Feb. 2012.

[7] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.

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

[9] H. K. Galoogahi, A. Fagg, and S. Lucey, "Learning background-aware correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1144–1152.

[10] H. K. Galoogahi, T. Sim, and S. Lucey, "Correlation filters with limited boundaries," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4630–4638.

[11] S. T. Birchfield and S. Rangarajan, "Spatiograms versus histograms for region-based tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 2. Jun. 2005, pp. 1158–1163.

[12] K. Nummiaro, E. Koller-Meier, and L. Van Gool, "Object tracking with an adaptive color-based particle filter," in *Pattern Recognition* (Lecture Notes in Computer Science), vol. 2449, no. 2. Berlin, Germany: Springer, 2002, pp. 353–360.

[13] H. Possegger, T. Mauthner, and H. Bischof, "In defense of color-based model-free tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2113–2120.

[14] T. Vojir, J. Noskova, and J. Matas, "Robust scale-adaptive mean-shift for tracking," *Pattern Recognit. Lett.*, vol. 49, pp. 250–258, Nov. 2014.

[15] Z. Zivkovic and B. Krose, "An EM-like algorithm for color-histogram-based object tracking," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, Jun./Jul. 2004, pp. 798–803.

[16] M. B. Kaaniche and F. Bremond, "Tracking hog descriptors for gesture recognition," in *Proc. 6th IEEE Int. Conf. Adv. Video Signal Based Surveill.*, Sep. 2009, pp. 140–145.

[17] H. Zhou, Y. Yuan, and C. Shi, "Object tracking using SIFT features and mean shift," *Comput. Vis. Image Understand.*, vol. 113, no. 3, pp. 345–352, Mar. 2009.

[18] Q. Wang, J. Fang, and Y. Yuan, "Multi-cue based tracking," *Neurocomputing*, vol. 131, pp. 227–236, May 2014.

[19] S. Avidan, "Support vector tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 8, pp. 1064–1072, Aug. 2004.

[20] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1619–1632, Aug. 2011.

[21] H. Grabner and H. Bischof, "On-line boosting and vision," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2006, pp. 260–267.

[22] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking," in *Computer Vision—ECCV*, vol. 5302. Berlin, Germany: Springer, 2008, pp. 234–247.

[23] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1409–1422, Jul. 2012.

[24] A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof, "On-line random forests," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Sep./Oct. 2009, pp. 1393–1400.

[25] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *Proc. Eur. Conf. Comput. Vis.*, vol. 7574, 2012, pp. 864–877.

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

[27] M. Danelljan, F. S. Khan, M. Felsberg, and J. van de Weijer, "Adaptive color attributes for real-time visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1090–1097.

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

[29] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ECO: Efficient convolution operators for tracking," in *Proc. 30th IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6931–6939.

[30] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Discriminative scale space tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1561–1575, Aug. 2017.

[31] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *Proc. Eur. Conf. Comput. Vis.*, vol. 9909, 2016, pp. 472–488.

[32] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Computer Vision—ECCV*, vol. 7575. Berlin, Germany: Springer, 2012, pp. 702–715.

[33] Y. Li and J. K. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *Proc. Comput. Vis.-ECCV Workshops*, vol. 8926, 2015, pp. 254–265.

[34] Q. Wang, M. Chen, F. Nie, and X. Li, "Detecting coherent groups in crowd scenes by multiview clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.

[35] S. Tian, X.-C. Yin, Y. Su, and H.-W. Hao, "A unified framework for tracking based text detection and recognition from Web videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 542–554, Mar. 2018.

[36] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional Siamese networks for object tracking," in *Proc. Comput. Vis.-ECCV Workshops*, vol. 9914, 2016, pp. 850–865.

[37] D. Held, S. Thrun, and S. Savarese, "Learning to track at 100 fps with deep regression networks," in *Computer Vision—ECCV*, vol. 9905. Berlin, Germany: Springer, 2016, pp. 749–765.

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

[39] H. Nam, M. Baek, and B. Han, "Modeling and propagating CNNs in a tree structure for visual tracking," 2016, *arXiv:1608.07242*. [Online]. Available: https://arxiv.org/abs/1608.07242

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

[41] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. S. Torr, "End-to-end representation learning for correlation filter based tracking," in *Proc. 30th IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5000–5008.

[42] S. A. Ahmadi and A. Mohammadzadeh, "A simple method for detecting and tracking vehicles and vessels from high resolution spaceborne videos," in *Proc. Joint Urban Remote Sens. Event (Jurse)*, 2017, pp. 1–4.

[43] B. Du, S. Cai, C. Wu, L. Zhang, and D. Tao, "Object tracking in satellite videos based on a multi-frame optical flow tracker," Apr. 2018, *arXiv:1804.09323*. [Online]. Available: https://arxiv.org/abs/1804.09323

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

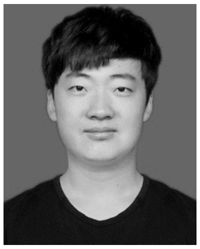
[45] R. M. Gray, *Toeplitz And Circulant Matrices: A Review (Foundations and Trends in Communications and Information Theory)*. Hanover, MA, USA: Now Publishers, 2006.

[46] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng. Trans.*, vol. 82, no. 1, pp. 35–45, 1960.

[47] Z. Kalal, K. Mikolajczyk, and J. Matas, "Forward-backward error: Automatic detection of tracking failures," in *Proc. 20th Int. Conf. Pattern Recognit. (ICPR)*, Oct. 2010, pp. 2756–2759.

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

[49] Y. Wu, J. Lim, and M. H. Yang, "Object tracking benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, Sep. 2015.

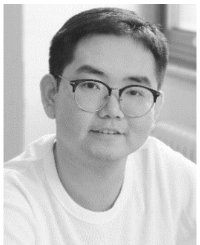


Shiyu Xuan received the B.Eng. degree in electronic and information engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2017. He is currently pursuing the M.S. degree in electronics and communication engineering with Technology and Engineering Center for Space Utilization, Chinese Academy of Sciences (CAS), Beijing, China.

His research interests include satellite video, unmanned aerial vehicle (UAV) video, and conventional video analysis, such as object tracking.

Shengyang Li received the B.Eng. degree in computer science and technology from the Shandong University of Science and Technology, Qingdao, China, in 2003, and the M.Eng. degree in remote sensing image processing and analysis from the Institute of Remote Sensing Applications, Chinese Academy of Sciences, Beijing, China, in 2006.

He is currently a Professor with the Technology and Engineering Center for Space Utilization, Chinese Academy of Sciences. His research interests include computer vision, target detection and tracking in video satellite, and remote sensing image analysis and understanding.



Mingfei Han received the B.Eng. degree in computer science and technology from Nankai University, Tianjin, China, in 2016, and the M.Eng. degree in computer technology from the University of Chinese Academy of Sciences, Beijing, China, in 2019. He is currently pursuing the Ph.D. degree in video analysis with the Faculty of Information Technology, Monash University, Melbourne, VIC, Australia, and CSIRO Data61, Eveleigh, NSW, Australia.

His research interests include remote sensing imagery and conventional video analysis, such as object detection and tracking, abnormal event detection, and action prediction.

Xue Wan received the B.Eng. and M.Eng. degrees in remote sensing from the School of Remote sensing and Information Engineering, Wuhan University, Wuhan, China, in 2010 and 2012, respectively, and the Ph.D. degree from Imperial College London, London, U.K., in 2015.

She is currently an Associate Professor with the Technology and Engineering Center for Space Utilization, Chinese Academy of Sciences, Beijing, China. Her research interests include remotely sensed image matching, change detection, vision-based navigation, and 3-D reconstruction.

Gui-Song Xia (SM'15) received the Ph.D. degree in image processing and computer vision from CNRS LTCI, Télécom ParisTech, Paris, France, in 2011.

From 2011 to 2012, he has been a Post-Doctoral Researcher with the Centre de Recherche en Mathmatiques de la Decision, CNRS, Paris Dauphine University, Paris, for one and a half years. He has also been a Visiting Scholar with the Department of Mathematics and Applications (DMA), École Normale Suprieure (ENS-Paris), Paris, since 2018. He is currently a Full Professor of computer vision and photogrammetry with Wuhan University, Wuhan, China. His research interests include mathematical modeling of images and videos, structure from motion, perceptual grouping, and remote sensing imaging.

Dr. Xia serves on the Editorial Boards for the journals *Pattern Recognition*, *Signal Processing: Image Communications*, and *EURASIP Journal on Image and Video Processing*.