# Vehicle Tracking on Satellite Video Based on Historical Model

Shili Chen, Taoyang Wang, Hongshuo Wang, Yunming Wang, Jianzhi Hong, Tiancheng Dong, and Zhen Li

***Abstract*—Vehicle tracking on satellite videos poses a challenge for the existing object tracking algorithms due to the few features, object occlusion, and similar objects appearance. To improve the performance of the object tracking algorithm, a historical-model-based tracker intended for satellite videos is proposed in this study. It updates the tracker by using the historical model of each frame in the video, which contains plenty of object information and background information, so as to improve tracking ability on few-feature objects. Furthermore, a historical model evaluation scheme is designed to obtain reliable historical models, which ensures that the tracker is sensitive to the object in the current frame, thus avoiding the impact caused by changes in object appearance and background. Besides, to solve the drift issue of the tracker caused by object occlusion and the appearance of similar objects, an antidrift tracker correction scheme is proposed as well. According to the comparative experiments conducted on satellite videos dataset SatSOT, our tracker produces an excellent performance. Moreover, sensitivity analysis, varying criteria comparative experiments, and ablation experiments are conducted to demonstrate that the proposed schemes are effective in improving the precision and success rate of the tracker.**

***Index Terms*—Correlation filter (CF), high-confidence tracking, motion estimation, object tracking, satellite video.**

## I. Introduction

OBJECT tracking represents an important subject of research on computer vision, the primary aim of which is to track the moving objects in the video and obtain the potential position of the object in each frame of the video through the computer. Due to the diversity of moving objects and the video equipment, it is necessary to adapt the object tracking algorithm to different data sources. For example, the ordinary video [1]–[3], the unmanned aerial vehicle (UAV) video [4], [5], the thermal infrared video [6], the synthetic aperture radar video [7], [8], and the satellite video [9]–[21], which is an emerging type of space-based video data in recent years. The satellite video has demonstrated such advantages as wide shooting range, high resolution, and the capability of continuously monitoring the target objects on land.

The object tracking on satellite videos has a wide range of applications in national defense, environmental protection, disaster prevention, and traffic monitoring [12], [22]. Our study focuses on the vehicle tracking, which is vital for traffic and military spying. Besides, due to the wide shooting range of satellite videos, the long-distance tracking is achievable, which is conducive to analyzing the driving motives and trajectories.

There are some issues when the vehicles are tracked on satellite videos. The first one is a small number of features and textures due to the small size of the object. Then, the vehicle is often occluded by trees, bridges, and other obstacles. Besides, similar objects often appear around the tracked object. Thus, it is a challenge to apply the existing object tracking algorithms to satellite videos.

According to the principle followed by the algorithm, the existing object tracking algorithms can be divided into two categories: generative models [23]–[26] and discriminative models [27]–[33]. Generative models focus on the characteristics of the object itself and track the object by conducting iterative search for the similar object frame by frame. Some classical tracking algorithms, such as MeanShift [23], CAMShift [24], particle filter [25], and optical flow [26] are classed as generative models. Differently, discriminative models focus on the difference between the object and the background. It takes background information into consideration, e.g., MOSSE [27], CSK [28], KCF [29], and STC [30]. With the widespread application of deep learning, the object tracking algorithms combined with deep learning have been gradually developed. Some algorithms extract deep convolutional features, e.g., C-COT [34] and ECO [35], both of which use the VGGNet [44], whereas others use end-to-end object tracking methods, e.g., CFNet [45], SiamFC [46], and MDNet [47]. Given the lack of background information, the generative algorithm only models the object itself, which is unreliable for the few-feature objects on satellite videos. By contrast, the discriminative algorithm introduces background information, which makes it advantageous over the generative

Manuscript received 30 May 2022; revised 19 July 2022; accepted 28 July 2022. Date of publication 1 August 2022; date of current version 16 September 2022. This work was supported in part by the Key Research and Development Program of the Ministry of Science and Technology under Grant 2018YFB0504905, in part by the High Resolution Earth Observation Systems National Science and Technology Major Projects under Grant GFZX0404130302, in part by the Civil Aerospace Advance Research Project under Grant D040107, and in part by the High Resolution Earth Observation Systems Industrialization Demonstration of Comprehensive Monitoring of Ecological Environment in Hubei under Grant 81-Y50G20-9001-22/23. *(Corresponding author: Taoyang Wang.)*

Shili Chen, Taoyang Wang, and Jianzhi Hong are with the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China (e-mail: sl_chen@whu.edu.cn; wangtaoyang@whu.edu.cn; hongjianzhi@qq.com).

Hongshuo Wang is with the Beijing Aerospace Automatic Control Institute, Beijing 100854, China (e-mail: hongshwang@163.com).

Yunming Wang and Tiancheng Dong are with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China (e-mail: wangyunming@whu.edu.cn; 2021106190028@whu.edu.cn).

Zhen Li is with the Institute of Remote Sensing Satellite, China Academy of Space Technology, Beijing 100094, China (e-mail: zhenli.irss@gmail.com).

Digital Object Identifier 10.1109/JSTARS.2022.3195522

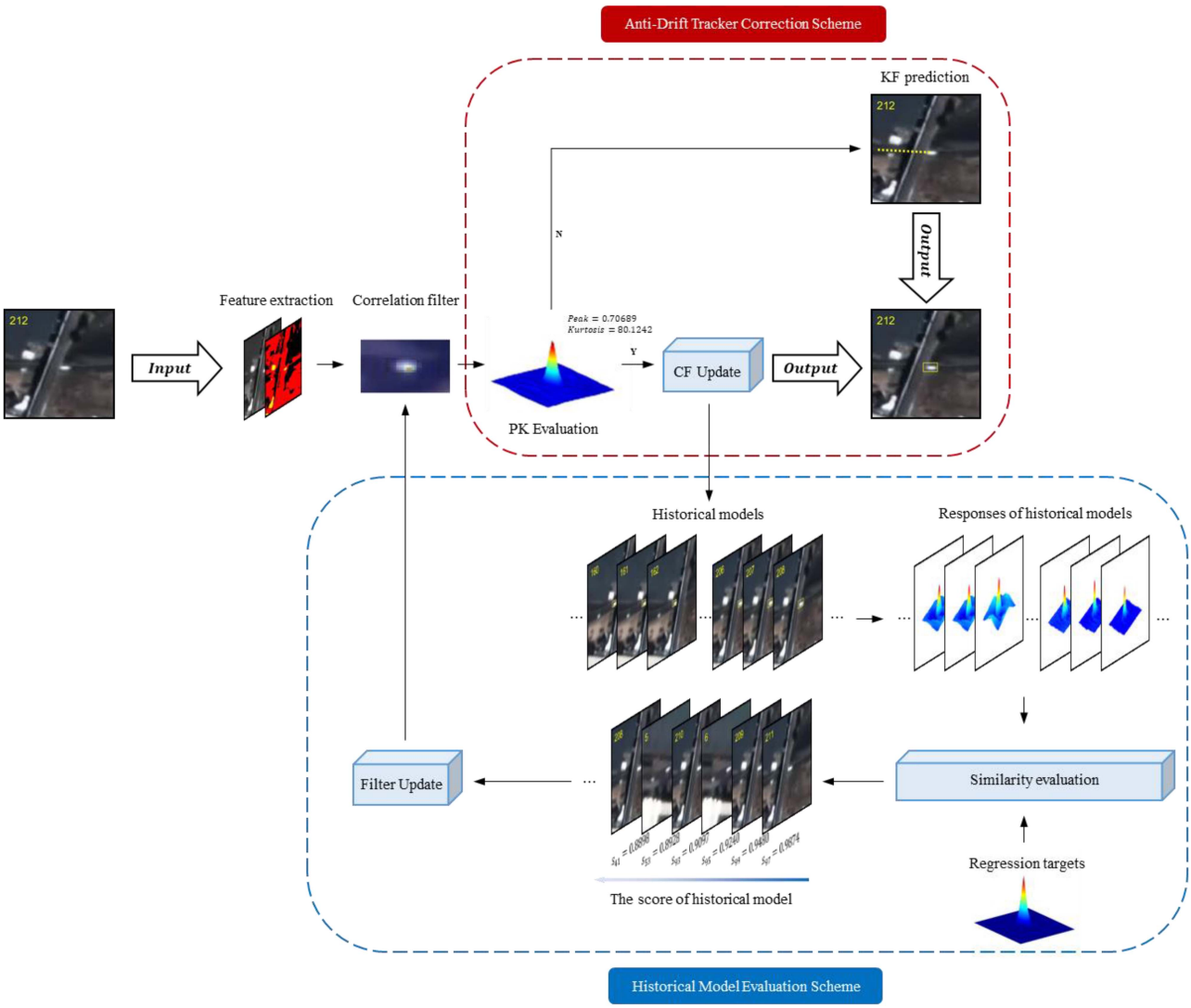



Fig. 1. Flow diagram of the proposed tracker HMTS.

algorithm. In particular, the correlation filter (CF), which is one of the discriminative algorithms, is widely used in object tracking on satellite videos due to its high speed and high precision. Although the algorithm based on deep learning is equally effective, it is disadvantaged by heavy computational workload and slow speed. In some cases, the few and weak features of objects make it inappropriate for neural networks to learn. In general, CF-based algorithms are superior to generative algorithms in accuracy and are superior to deep-learning-based algorithms in speed, which makes them have greater advantages on the satellite videos with larger image sizes and fewer features. For this reason, the CF is applied in this study.

For most of the current CF algorithms, the focus is on ridge regression. By learning the features of the region of interest (ROI), they can classify the object and background to locate the object. When the filter is updated, most algorithms use a single ROI in the current frame and ignore the historical information of the previous frames, which leads to the waste of information. Nevertheless, historical information contains a great deal of the object information and background information required for modeling the object. Especially, background information could be inputted into CF as negative samples, which could improve the tracking performance for the few-features object. For this reason, a historical-model-based tracker for satellite (HMTS) videos is proposed in this study, as illustrated in the flow diagram (see Fig. 1). Historical models (HMs) are applied to update the tracker. However, not all HMs are suitable due to changes in object appearance and background during the tracking process. Hence, a scheme is used to find the HMs that are best suited to the current frame. In addition, since the vehicle is often occluded by obstacles or there are similar objects appearing around the vehicle, a strategy is adopted to detect vehicle state and prevent the drift of the tracker. The peak and kurtosis (PK) of the response map of the CF [50] are employed to monitor the vehicle state. Additionally, Kalman filter (KF) is adopted to correct the tracking result to avoid drift. To sum up, the contributions of this study are as follows:

1) In this article, we designed a novel HMTS videos to improve the tracking performance on small-size objects, thus addressing the trace failure caused by a lack of features.

It improves the acquisition of object information by utilizing historical information, ensures the reliability of tracking, and reduces historical information waste.

2) We proposed an HM evaluation scheme to avoid the impact caused by changes in object appearance and background. The scheme relies on cross-correlation functions to measure the similarity between the response map and the ideal regression targets. HMs are scored by this scheme to identify the best-suited ones for the current frame.
3) We constructed a tracker correction scheme to prevent the tracker drift caused by object occlusion and similar objects appearance. The scheme uses KF to estimate the motion of the object and PK for the response map evaluation. Based on PK, the trajectory of the tracker is corrected by the prediction function of KF.

The rest of this article is organized as follows. In Section II, the relevant work is summarized. Section III presents the design of the proposed tracker HMTS. In Section IV, the experiments are detailed, including configurations, experimental design, experimental analysis, and experimental results. Finally, Section V concludes this article.

## II. Related Literature

### A. CF-Based Object Tracking

The emergence of CF marks a key milestone in the development of object tracking algorithm. By introducing the Fourier transform method to convert the originally complex matrix operations into calculations in the frequency domain, it accelerates the calculation process. At the same time, CF increases the use of background information, which improves the accuracy of tracking. In the earliest CF algorithm MOSSE [27], the least squares method is used to distinguish between the background and the object. Then, kernel CF is used by CSK [28] and KCF [29] to introduce circulant matrix, which increases the number of samples, thus improving the accuracy of training. Following the KCF, scholars have successively improved the KCF framework to develop many algorithms based on CF. There are three main approaches to improving the CF. The first one is to apply a suitable feature extraction algorithm [34]–[37]. In [36], color names (CNs) are used, whereas in [34], deep convolutional features are used. The second one is to adapt to the object scale change by introducing a scale-adaptive mechanism [38], [39]. The mechanism is effective in increasing the success rate of the tracker. The last one is to improve the structure of the filter [40]–[43], e.g., BACF [41] adds the spatial regularization term, and AutoTrack [43] adds the temporal regularization term.

### B. Object Tracking on Satellite Video

The existing object tracking algorithms have been widely used for the tracking of objects on satellite videos. The traditional generative model is applied in [9] and [10]. Besides, CF algorithm is used in [11]–[17]. Moreover, deep learning is adopted in [18]–[21]. At present, traditional generative model is rarely adopted for satellite videos, which is attributed mainly to its low accuracy and slow speed. By contrast, the discriminative model has been more commonly used. A major solution is to improve CF, which can be achieved from three perspectives. The first one is to change the feature extraction algorithm, the second one is to use motion model, and the last one is to introduce tracker state monitoring scheme. To change the feature extraction algorithm, Shao et al. [12] combined LK optical flow method and histogram of oriented gradient (HOG) feature extraction algorithm to extract features, whereas Wu et al. [15] combined Hu CF and median filter to extract features. Due to the rotation invariance of Hu invariant moment, this algorithm performs well in tracking rotation objects. In [17], Gabor filter is employed to extract features, which leads to an excellent performance for the objects with textures. In terms of motion models, KF is frequently used to predict the position of the moving object [14]–[16], which improves the robustness of the algorithm. As for the tracker state monitoring scheme, Wang et al. [17] proposed the tracking status monitoring indicators (TCMIs) based on the Bayesian framework. If the object is occluded, TCMI can be relied on to guide the tracker, with updating terminated to prevent model drift. In addition, Xuan et al. [16] used the peak value of response map to monitor tracker states and guide the updating of the tracker.

### C. Benchmark Dataset Based on Satellite Video

In addition to the improvement of algorithm, the benchmark dataset based on satellite video has also been proposed to make the verification of algorithms more reliable. Yin et al. [48] constructed a large-scale satellite video dataset with a wide range of annotations. At the same time, a benchmark was proposed to evaluate algorithms for their performance, e.g., the multiobject tracking algorithm and the single-object tracking algorithm on satellite videos. Zhao et al. [49] put forward another densely annotated satellite video dataset, which is purposed to evaluate the single-object tracking algorithm.

## III. Proposed Method

In this section, it will be explained how to apply the HM, cross-correlation function, PK, and KF for CF-based object tracking. First, a tracker HMTS based on history model is proposed to track the object. Then, an HM evaluation scheme is introduced to validate each HM, so as to choose the HMs that are sensitive to the object in the current frame. Finally, a scheme is introduced to correct the tracker by using KF and PK.

### A. Overall Architecture

In our tracker, kernel CF (KCF) [29] is taken as the baseline. Suppose there is a vectorized feature map $\boldsymbol{X}$, $\boldsymbol{X} \in \mathbb{R}^D$. KCF solves the following ridge regression problem:

$$\min_{\boldsymbol{w}} \sum_{i=1}^{D} \left(\boldsymbol{w}^T \boldsymbol{X}(i) - \boldsymbol{y}(i)\right)^2 + \lambda \boldsymbol{w}_2^2 \tag{1}$$

where $\boldsymbol{w}$ represents a filter; $\boldsymbol{X}(i)$ refers to the $i$th circular shift of $\boldsymbol{X}$; $\boldsymbol{y}(i)$ indicates the regression target for $\boldsymbol{X}(i)$; $\lambda \boldsymbol{w}_2^2$ denotes a shrinkage penalty, which controls overfitting; $\lambda$ is referred

to as the penalty parameter; and the superscript $^T$ refers to transposition.

By mapping the linear inputs $\boldsymbol{X}(i)$ to nonlinear space $\varphi(\boldsymbol{X}(i))$, the filter $\boldsymbol{w}$ can be expressed as a linear combination of $\varphi(\boldsymbol{X}(i))$. In nonlinear space, as opposed to linear space, the feature map can be simply split into the object and background. Additionally, the kernel trick can be used to express $\boldsymbol{w}^T\boldsymbol{X}(i)$ as

$$\begin{aligned}\boldsymbol{w}^T X\left(i\right) &= \sum_{J=1}^{D}\alpha\left(j\right)\varphi^T\left(\boldsymbol{X}\left(j\right)\right)\varphi\left(\boldsymbol{X}\left(i\right)\right)\\ &= \sum_{J=1}^{D}\alpha\left(j\right)\mathcal{K}\left(\varphi\left(\boldsymbol{X}\left(j\right)\right),\varphi\left(\boldsymbol{X}\left(i\right)\right)\right)\\ &= \boldsymbol{K_{XX}}\ \left(i\right)\alpha \end{aligned} \tag{2}$$

where $\mathcal{K}(\cdot)$ refers to the kernel function (e.g., radial basis function) used to compute the dot-product of nonlinear space; $\boldsymbol{K_{XX}}$ represents the kernel matrix with elements $\boldsymbol{K_{XX}}\left(i,j\right) = \mathcal{K}(\varphi(\boldsymbol{X}(j)),\varphi(\boldsymbol{X}(i)))$; the $i$th rows of $\boldsymbol{K_{XX}}$are referred as $\boldsymbol{K_{XX}}\left(i\right) = [\boldsymbol{K_{XX}}(i,1),\ \cdots,\boldsymbol{K_{XX}}(i,D)]$ ; and $\boldsymbol{\alpha}$ is the vector of coefficients $\alpha(i)$ $(i\ =\ 1,\cdots,D)$. Similarly, $\boldsymbol{w}_2^2$ can be expressed as

$$\boldsymbol{w}_2^2 = \boldsymbol{\alpha}^T\ \boldsymbol{K_{XX}}\alpha\,. \tag{3}$$

Thus, (1) can be reformulated as

$$\min_{\boldsymbol{\alpha}}\ \sum_{i\ =\ 1}^{D}\left(\boldsymbol{K_{XX}}\left(i\right)\boldsymbol{\alpha}-\boldsymbol{y}\left(i\right)\right)^2+\lambda\boldsymbol{\alpha}^T\boldsymbol{K_{XX}}\boldsymbol{\alpha}\,. \tag{4}$$

The solution of (4) is expressed as

$$\begin{aligned}\alpha =&\ \arg\min_{\boldsymbol{\alpha}}\left\{\boldsymbol{K_{XX}}\boldsymbol{\alpha}-\boldsymbol{y}_2^2+\lambda\boldsymbol{\alpha}^T\boldsymbol{K_{XX}}\boldsymbol{\alpha}\right\}\\ &=\left(\boldsymbol{K_{XX}}+\lambda\boldsymbol{I}\right)^{-1}y\end{aligned} \tag{5}$$

where $\boldsymbol{y}$ is defined as $\boldsymbol{y}\ =[\boldsymbol{y}(1),\cdots,\boldsymbol{y}(D)]^{\boldsymbol{T}}$ . $\boldsymbol{I}$ refers to a $D\times D$ identity matrix. The filter $\boldsymbol{w}$ is replaced by $\boldsymbol{\alpha}$. $\boldsymbol{K_{XX}}+\lambda\boldsymbol{I}$ can be reformulated as circulant matrix $C(K_{\boldsymbol{XX}}+\lambda\boldsymbol{\delta})(\delta\ =[1,0,\cdots,0]\ )$. $K_{\boldsymbol{XX}}$ refers to the first row of $\boldsymbol{K_{XX}}$. By using the discrete Fourier transform (DFT), (5) can be expressed as

$$\hat{\boldsymbol{\alpha}} = \frac{\hat{\boldsymbol{y}}}{\hat{K}_{\boldsymbol{XX}}+\boldsymbol{\lambda}} \tag{6}$$

where the hat $\widehat{\ }$ denotes the DFT of a vector. When KCF filter is used, the filter $\hat{\boldsymbol{\alpha}}_{T-1}$ is updated by (6) with $\hat{K}_{\boldsymbol{z}_{T-1}\boldsymbol{z}_{T-1}}$. $\boldsymbol{z}_{T-1}$ refers to a target-centered patch of last frame $T-1$ whose target is located in the center. Suppose that a new patch $\boldsymbol{x}_T$ is sampled in the current frame $T$, and the location of the target is detected by

$$\boldsymbol{R}_T = \mathcal{F}^{-1}\ \left(\hat{K}_{\bar{\boldsymbol{z}}_T\boldsymbol{x}_T}\hat{\boldsymbol{\alpha}}_{T-1}\right) \tag{7}$$

where $\mathcal{F}^{-1}$ denotes the IDFT operator. $\hat{K}_{\bar{\boldsymbol{z}}_T\boldsymbol{x}_T}$ is computed by $\boldsymbol{x}_T$ and sample model $\bar{\boldsymbol{z}}_T$. The position of the target is determined by the position of the maximum value of $\boldsymbol{R}_T$.

Typically, the classical KCF does not use the filter $\hat{\boldsymbol{\alpha}}$ derived from (6) directly when an object is detected. Instead, linear interpolation is performed to update the filter $\overline{\hat{\boldsymbol{\alpha}}}_T$ as follows:

$$\begin{aligned}\overline{\hat{\boldsymbol{\alpha}}}_T =&\ \beta\hat{\boldsymbol{\alpha}}_{T-1}+\left(1-\beta\right)\overline{\hat{\boldsymbol{\alpha}}}_{T-1}\\ &=\beta\sum_{i=0}^{T-3}\left(1-\beta\right)^i\hat{\boldsymbol{\alpha}}_{T-1-i}+\left(1-\beta\right)^{T-2}\hat{\boldsymbol{\alpha}}_1\end{aligned} \tag{8}$$

where $\beta$ represents the coefficient of interpolation. Similarly, the sample model $\bar{\boldsymbol{z}}_T$ is also derived from linear interpolation

$$\begin{aligned}\bar{\boldsymbol{z}}_T =&\ \gamma\boldsymbol{z}_{T-1}+\left(1-\gamma\right)\bar{\boldsymbol{z}}_{T-1}\\ &=\gamma\sum_{i=0}^{T-3}\left(1-\gamma\right)^i\boldsymbol{z}_{T-1-i}+\left(1-\gamma\right)^{T-2}\boldsymbol{z}_1\end{aligned} \tag{9}$$

where $\gamma$ is also referred to as the coefficient of interpolation. By using (8) and (9), the tracker can be updated at a fixed learning rate.

Updating the filter with a fixed proportion can make the filter retain part of the historical information and avoid the interference from new information. However, the fixed learning rate also causes the tracker to treat all the historical information indiscriminately, which causes the waste of historical information. Meanwhile, since the initial value of the tracker is obtained from the first frame, the information contained in the first frame will be exponentially attenuated when the tracker is updated [see (8) and (9)]. In case of a low learning rate, $\hat{\boldsymbol{\alpha}}_1$ and $\boldsymbol{z}_1$ may account for a larger proportion in the filter and sample model, respectively, in the early stage of tracking, which has a significant impact on the tracker. Therefore, the tracker is capable to recognize the object, which is similar to the first frame. Hence, when the feature of the object or the background changes, the tracking may fail.

To make full use of historical information, our tracker retains the filter and the target-centered patch for each frame as HMs $H\ =\{(\hat{\boldsymbol{\alpha}}_i,\boldsymbol{z}_i,s_i)\}_{i=1}^{T-1}$ , where $s_i$ represents a score of each HM. As for the approach to score learning, it will be detailed in Section III-B. Suppose that a new patch $\boldsymbol{x}_T$ is sampled in the current frame $T$, then the filter $\overline{\hat{\boldsymbol{\alpha}}}_T$ is calculated by

$$\begin{aligned}\overline{\hat{\boldsymbol{\alpha}}}_T &= \beta\dot{\hat{\boldsymbol{\alpha}}}_{T-1}+\left(1-\beta\right)\overline{\hat{\boldsymbol{\alpha}}}_{T-1}\\ \dot{\hat{\boldsymbol{\alpha}}}_{T-1} &= \frac{\sum_{i=1}^{T-1}s_i\hat{\boldsymbol{\alpha}}_i}{\sum_{i=1}^{T-1}s_i}.\end{aligned} \tag{10}$$

The sample model $\bar{\boldsymbol{z}}_T$ is calculated by

$$\begin{aligned}\bar{\boldsymbol{z}}_T =&\ \gamma\dot{\boldsymbol{z}}_{T-1}+\left(1-\gamma\right)\bar{\boldsymbol{z}}_{T-1}\\ \dot{\boldsymbol{z}}_{T-1} &= \frac{\sum_{i=1}^{T-1}s_i\boldsymbol{z}_i}{\sum_{i=1}^{T-1}s_i}\end{aligned} \tag{11}$$

where linear interpolation is still performed to update our filter and sample model at the learning rate $\beta$ and $\gamma$, respectively. At each frame, $\dot{\hat{\boldsymbol{\alpha}}}_{T-1}$ and $\dot{\boldsymbol{z}}_{T-1}$ are calculated. They both consist of the HMs. The score $s_i$ $(i\ =\ 1,\cdots,T-1)$ leads to $\hat{\boldsymbol{\alpha}}_i$ $(i\ =\ 1,\cdots,T-1)$, which is more sensitive to $\boldsymbol{x}_T$ accounting for a larger proportion in $\dot{\hat{\boldsymbol{\alpha}}}_{T-1}$. In the same way, $s_i$ $(i\ =\ 1,\cdots,T-1)$ leads to $\boldsymbol{z}_i$ $(i\ =\ 1,\cdots,T-1)$, which is more similar as $\boldsymbol{x}_T$ accounts for a larger proportion in $\dot{\boldsymbol{z}}_{T-1}$. In these operations, $\dot{\hat{\boldsymbol{\alpha}}}_{T-1}$ and $\dot{\boldsymbol{z}}_{T-1}$ classify the object and background in $\boldsymbol{x}_T$ with high precision. In the meantime,

**Algorithm 1:** HMTS Tracking Algorithm.

**Input**: Current Image $I_t$, Previous Position $p_{t-1}$, Historical Models $H = \{(\widehat{\boldsymbol{\alpha}}_i, \boldsymbol{z}_i)\}_{i=1}^{t-1}$, Previous Filter $\overline{\widehat{\boldsymbol{\alpha}}}_{t-1}$, Previous Sample Model $\overline{\boldsymbol{z}}_{t-1}$, History Peak $Peak\_set$, Historical Kurtosis $BK\_set$, Regression Targets $\boldsymbol{y}$.

**Output:** The Object Position $p_t$, Updated Historical Models $H = \{(\widehat{\boldsymbol{\alpha}}_i, \boldsymbol{z}_i)\}_{i=1}^{t}$, Updated Filter $\overline{\widehat{\boldsymbol{\alpha}}}_t$, Updated Sample Model $\overline{\boldsymbol{z}}_t$, Updated history Peak $Peak\_set$, Updated historical Kurtosis $BK\_set$,

**Repeat:**

Extract the image patch's feature $\boldsymbol{x}_t$ from the image $I_t$.

**if** not the first frame

- Calculate the response map $\boldsymbol{R}_t$ using the Previous Filter $\overline{\widehat{\boldsymbol{\alpha}}}_{t-1}$ and Previous Sample Model $\overline{\boldsymbol{z}}_{t-1}$ (Eq. 12).
- Find the detected potion $d_t$ by searching the maximum position of response map.

**end if**

Extract the target-centered patch's feature $\boldsymbol{z}_t$.

Update CF Filter $\widehat{\boldsymbol{\alpha}}_t$ (Eq. 6).

Update HMs $H = \{(\widehat{\boldsymbol{\alpha}}_i, \boldsymbol{z}_i)\}_{i=1}^{t}$.

**if** not the first frame

- Calculate Peak $max(\boldsymbol{R}_t)$ of response map $\boldsymbol{R}_t$.
- Update History Peak $Peak\_set$.
- Calculate Peak threshold $P_{tr}$ (Eq. 22).
- Calculate Kurtosis $BK_i$ of response map $\boldsymbol{R}_t$.
- Update Historical Kurtosis $BK\_set$.
- Calculate Kurtosis threshold $K_{tr}$ (Eq. 22).
- **If** $max(\boldsymbol{R}_t) > P_{tr}$ and $BK_i > K_{tr}$
  - Calculate response $\boldsymbol{R}_i$ of HM on $\boldsymbol{z}_t$ (Eq. 15).
  - Calculate score $s_i$ of HM (Eq. 16 and Eq. 17).
  - Update Filter $\overline{\widehat{\boldsymbol{\alpha}}}_t$ (Eq. 10).
  - Update Sample Model $\overline{\boldsymbol{z}}_t$ (Eq. 11).
  - Update the object position $p_t = d_t$
- **else**
  - Predict Position $\bar{\hat{p}}_t$ by KF (Eq. 21).
  - Update Filter $\overline{\widehat{\boldsymbol{\alpha}}}_t = \overline{\widehat{\boldsymbol{\alpha}}}_{t-1}$.
  - Update Sample Model $\overline{\boldsymbol{z}}_t = \overline{\boldsymbol{z}}_{t-1}$.
  - Update the object position $p_t = \bar{\hat{p}}_t$.
- **end if**

**else**

- Initialize Filter $\overline{\widehat{\boldsymbol{\alpha}}}_t = \widehat{\boldsymbol{\alpha}}_t$.
- Initialize Sample Model $\overline{\boldsymbol{z}}_t = \boldsymbol{z}_t$.

**end if**

**Until** the end of video sequence.

it makes full use of the HMs, with the historical information treated differently. In addition, when the feature of the object or background changes, the scores can be used to reduce the impact of the initial value by adjusting the proportion of the initial value dynamically, because the scores will change in each frame. After the filter $\overline{\widehat{\alpha}}_T$ and the sample model $\bar{\boldsymbol{z}}_T$ are obtained, the response of $\boldsymbol{x}_T$ can be calculated by

$$\boldsymbol{R}_T = \mathcal{F}^{-1}\left(\hat{K}_{\bar{\boldsymbol{z}}_T \boldsymbol{x}_T} \overline{\widehat{\boldsymbol{\alpha}}}_T\right). \tag{12}$$

The position of the maximum value of $\boldsymbol{R}_T$ is that of the object.

### B. HM Evaluation Scheme

In Section III-A, it is proposed to update the tracker by using HMs. In order to improve the capability of the tracker to distinguish between the object and the background, it is necessary to improve the sensitivity of the selected filters to the object. Meanwhile, the selected sample models must be similar to the object. In this section, an HM evaluation scheme will be introduced to obtain the appropriate filters and sample models by scoring the HM.

The cross-correlation function is applied to measure the correlation of two signals in signal analysis. The discrete cross-correlation function is expressed as follows:

$$S(\tau) = \sum_{n=-\infty}^{+\infty} \phi_1(n)\, \phi_2(n+\tau) \quad (\tau \in (-\infty, +\infty)) \tag{13}$$

where $\phi_1$ and $\phi_2$ represent signals. $\boldsymbol{S}(\tau)$ is referred to as the similarity of two signals when one of them is applied to a $\tau$-step discrete circular shift. The higher the value of $\boldsymbol{S}(\tau)$, the higher the similarity of the two signals. The maximum value of $\boldsymbol{S}(\tau)$ $(\tau \in (-\infty, +\infty))$ can be taken to represent global similarity. Meanwhile, the DFT is used to define the discrete cross-correlation function as

$$S = \mathcal{F}^{-1}\left(\hat{\phi}_1^H \hat{\phi}_2\right) \tag{14}$$

where $\boldsymbol{S}$ represents the vector of $\boldsymbol{S}(\tau)$ $(\tau \in (-\infty, +\infty))$ and the superscript $^H$ refers to conjugate transposition.

By taking the regression targets and the response as signals, the global similarity of them can be calculated. In theory, the better the performance of the tracker, the higher the global similarity between the response and the regression targets. According to this theory, an HM evaluation scheme is designed to score the HMs.

In frame $T$, after position detection is completed, a new target-centered patch $\boldsymbol{z}_T$ can be obtained and a new filter $\widehat{\boldsymbol{\alpha}}_T$ can be calculated by (6). HMs are updated to $H = \{(\widehat{\boldsymbol{\alpha}}_i, \boldsymbol{z}_i, s_i)\}_{i=1}^{T}$. In order to obtain $s_i$ $(i = 1, \cdots, T)$, each HM is combined as a tracker in the first place. Subsequently, the responses can be received on $\boldsymbol{z}_T$

$$\boldsymbol{R}_i = \mathcal{F}^{-1}\left(\hat{K}_{\boldsymbol{z}_i \boldsymbol{z}_T} \widehat{\boldsymbol{\alpha}}_i\right). \tag{15}$$

Then, the discrete cross-correlation function is applied on $\boldsymbol{R}_i$ $(i = 1, \cdots, T)$ and regression targets $\boldsymbol{y}$

$$\boldsymbol{S}_i = \mathcal{F}^{-1}\left(\boldsymbol{y}^H \boldsymbol{R}_i\right). \tag{16}$$

The score $s_i$ $(i = 1, \cdots, T)$ for the $i$th HM in frame $T$ is given by

$$s_i = \begin{cases} \max(\boldsymbol{S}_i), & \text{Rank}(\max(\boldsymbol{S}_i)) \le \theta \\ 0, & \text{Rank}(\max(\boldsymbol{S}_i)) > \theta \end{cases} \tag{17}$$

where $\max(\cdot)$ denotes the maximum of the vector; $\text{Rank}(\max(\boldsymbol{S}_i))$ indicates the index of $\max(\boldsymbol{S}_i)$ in the set $\{\max(\boldsymbol{S}_i)\}_{i=1}^{T}$, which is ranked in descending order; and $\theta$ refers to the threshold.

By using this scheme, the HM, which performs better in distinguishing between the object and background in $\boldsymbol{z}_T$, is

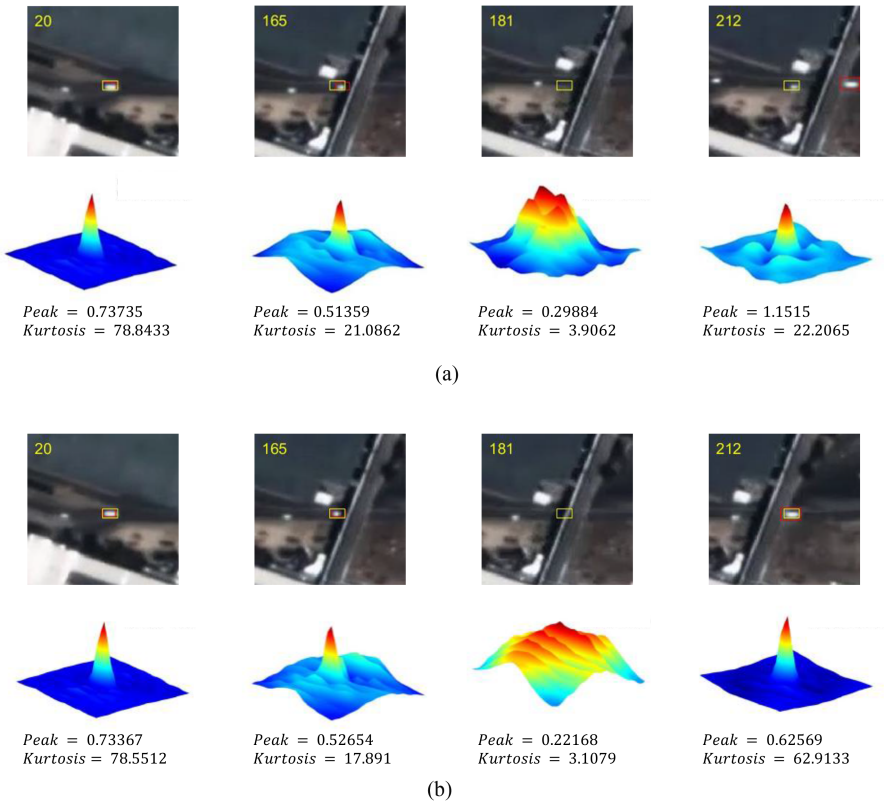



Fig. 2. Illustration of tracking result and response map. The yellow box is tracking box and the red box is ground truth. The object is obscured at frame 181. (a) Tracking without antidrift tracker correction scheme. It shows the performance of tracker without antidrift tracker correction scheme. (b) Tracking with antidrift tracker correction scheme. It shows the performance of tracker with antidrift tracker correction scheme.

given a higher score. Meanwhile, since the HM with low global similarity is irrelevant to confirming the object location, the scores of this kind of HMs are set to 0. All scores are used to update the tracker [see (10) and (11)].

Under this scheme, the historical information can be applied flexibly. Besides, the filter and the sample model are made more robust for object tracking.

### C. Antidrift Tracker Correction Scheme

It is a challenge to prevent tracker drift for object tracking. In general, tracker drift occurs when object occlusion happens and similar objects appear. In these cases, a significant deviation can arise between the maximum position of the response map and the ground truth of the object. Besides, the response map fluctuates significantly or shows multiple peaks. Therefore, it is necessary to monitor the tracking state and correct trajectory in the process of tracking. In this section, it will be explained how the KF and the PK of the response map can be applied to correct the tracker.

Assume that the ground truth of the vehicle is represented by $\{p_t\}_{t=1}^{T}$. The motion of the vehicle can be expressed as dynamic equation $p_t = f_t(p_{t-1}, u_t)$. In the process of tracking, the position detected by our tracker can be expressed as $\{d_t\}_{t=1}^{T}$. The relationship between detected position and ground truth is expressed as measurement equation $d_t = h_t(p_t, v_t)$, where both $v_t$ and $u_t$ are the noises that are independent and uniformly distributed. If $u_t$ and $v_t$ are Gaussian while dynamic equation and measurement equation are linear, KF is applicable to estimate the motion of the vehicle [51].

The dynamic equation and measurement equation of KF are expressed as

$$p_t = f_t\ (p_{t-1}, u_t) = Ap_{t-1} + u_t \tag{18}$$

$$d_t = h_t\ (p_t, v_t) = Bp_t + v_t \tag{19}$$

where $A$ denotes the state transition matrix and $B$ denotes the measurement matrix. The aim of KF is two-fold. One is correction and the other is prediction. Correction means correcting the measured position, in our case is the detected position $d_t$, to an estimate of the ground truth $\hat{p}_t$. Prediction means using $\hat{p}_t$ to predict the ground truth $\bar{\hat{p}}_{t+1}$ in the next frame. Herein, the hat $\hat{}$ denotes the estimated value, and the bar $\bar{}$ denotes the predicted value. The way of correction is defined as

$$\begin{cases} G_t = \bar{C}_t\, B^T \left(B\bar{C}_t B^T + V_t\right)^{-1} \\ \hat{p}_t = \bar{\hat{p}}_t + G_t\left(d_t - B\bar{\hat{p}}_t\right) \\ C_t = (I - G_t B)\ \bar{C}_t \end{cases}. \tag{20}$$

The way of prediction is defined as

$$\begin{cases} \bar{\hat{p}}_{t+1} = A\hat{p}_t \\ \bar{C}_{t+1} = AC_tA^T + U_t \end{cases} \tag{21}$$

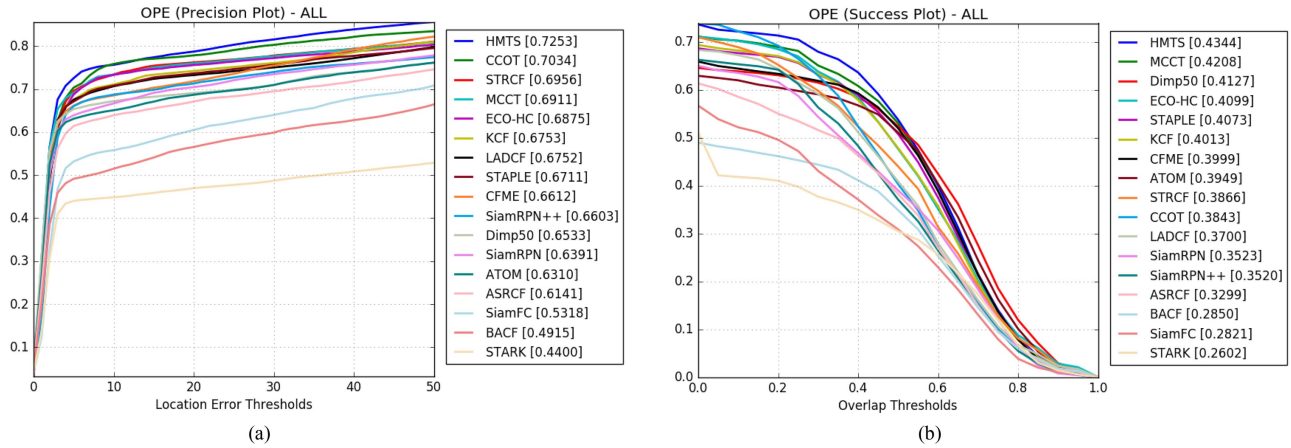



Fig. 3. Overall OPE results on SatSOT. The legend shows the sorting of algorithms based on five pixels location error threshold and AOE, respectively. (a) Precision plot. (b) Success plot.

where $G_t$ denotes the Kalman gain; $C_t$ indicates the covariance matrix of error between ground truth and estimation $(p_t - \hat{p}_t)$; $\bar{C}_t$ represents the covariance matrix of error between ground truth and prediction $(p_t - \overline{\hat{p}}_t)$; $V_t$ refers to the covariance matrix of $v_t$; $U_t$ stands for the covariance matrix of $u_t$; and $I$ denotes an identity matrix.

In the current frame, the detected position $d_t$ is inputted into the KF to obtain a predicted position $\overline{\hat{p}}_{t+1}$ for the next frame. If the tracker is found to have drifted in the next frame, the detected position $d_{t+1}$ will be replaced by the predicted position $\overline{\hat{p}}_{t+1}$. To measure the reliability of tracker, the high-confidence criterion proposed by Han et al. is adopted [50]. It relies on PK of the response map to detect whether the tracker has drifted. The two measurement thresholds are defined as

$$\begin{cases} P_{tr} = \delta_1 \times \dfrac{\sum_{i=1}^{T} \max(\boldsymbol{R}_i)}{T} \\ K_{tr} = \delta_2 \times \dfrac{\sum_{i=1}^{T} BK_i}{T} \end{cases} \tag{22}$$

where $P_{tr}$ and $K_{tr}$ represent peak threshold and kurtosis threshold derived from the historical average values with certain ratio $\delta_1$ and $\delta_1$, respectively; $\max(\boldsymbol{R}_i)$ refers to the maximum of response that denotes the peak value; and $BK_i$ denotes the kurtosis value. Ideally, the response map resembles a Gaussian distribution, with a single peak and smooth surroundings. When the drift occurs, the response map exhibits multiple peaks and fluctuates significantly, with the PK in decline (see Fig. 2). Therefore, if the PK of the response map at the current frame reaches above the PK threshold, it can be considered that the response map at the current frame is reliable and no drift occurs to the tracker. If one indicator falls below the threshold, the position detected by the tracker is unreliable and it is necessary to replace it with the position predicted by KF. Meanwhile, the correction of KF and the update of tracker will be terminated until the tracker returns to normal. Fig. 2 shows the effect of antidrift tracker correction scheme. When PK decreases, the tracker with the scheme ceases to be updated and KF predicts the position of the object. Since the incorrect information is not accepted by tracker, the tracking returns to normal when the object is no longer obscured. On the contrary, the tracker without the scheme keeps updating and cannot find the position of the object. When the object appears again, the tracker becomes ineffective.

This scheme is effective in avoiding tracker drift through the combination of KF and PK. Under the strategy of updating termination, the tracker and KF avoid receiving false information, which ensures the reliability of the tracker and the accuracy of KF. When the tracker stops working, the prediction function of KF can be used to predict the trajectory reasonably and ensure the continuity of the trajectory. The full algorithm is detailed in Algorithm I.

## IV. Experimental Results and Analysis

### A. Experimental Setups

*1) SatSOT Dataset:* As a densely annotated satellite video single-object tracking benchmark dataset, the SatSOT dataset [49] covers four categories of objects, including vehicle, train, plane, and ship. Herein, the videos of vehicle are used, involving 65 objects, 19 948 frames, and 9 challenges. The object is annotated by a bounding box and recorded with upper-left coordinate, the width, and height of the bounding box. Table I lists the description of nine challenging attributes given by the article [49].

*2) Evaluation Metric:* The evaluation metric used in the experiments is one-pass evaluation (OPE) [1], [2], with two evaluation indicators involved to perform quantitative evaluation, i.e., precision and successful rate. By plotting precision plot and success plot, the performance of algorithms can be visualized.

Precision is an indicator used to evaluate the error between the detected position of the tracker and the ground truth. The location error of the tracker can be obtained by calculating the Euclidean distance between the center point of the tracking box and the center point of the ground truth. Typically, there is a threshold of location error (e.g., 20), and the percentage of frames whose location error is smaller than this threshold is referred to as the tracker's precision. In the meantime, precision plot is created. The horizontal axis of the precision plot is the location error threshold, and the vertical axis is the precision.

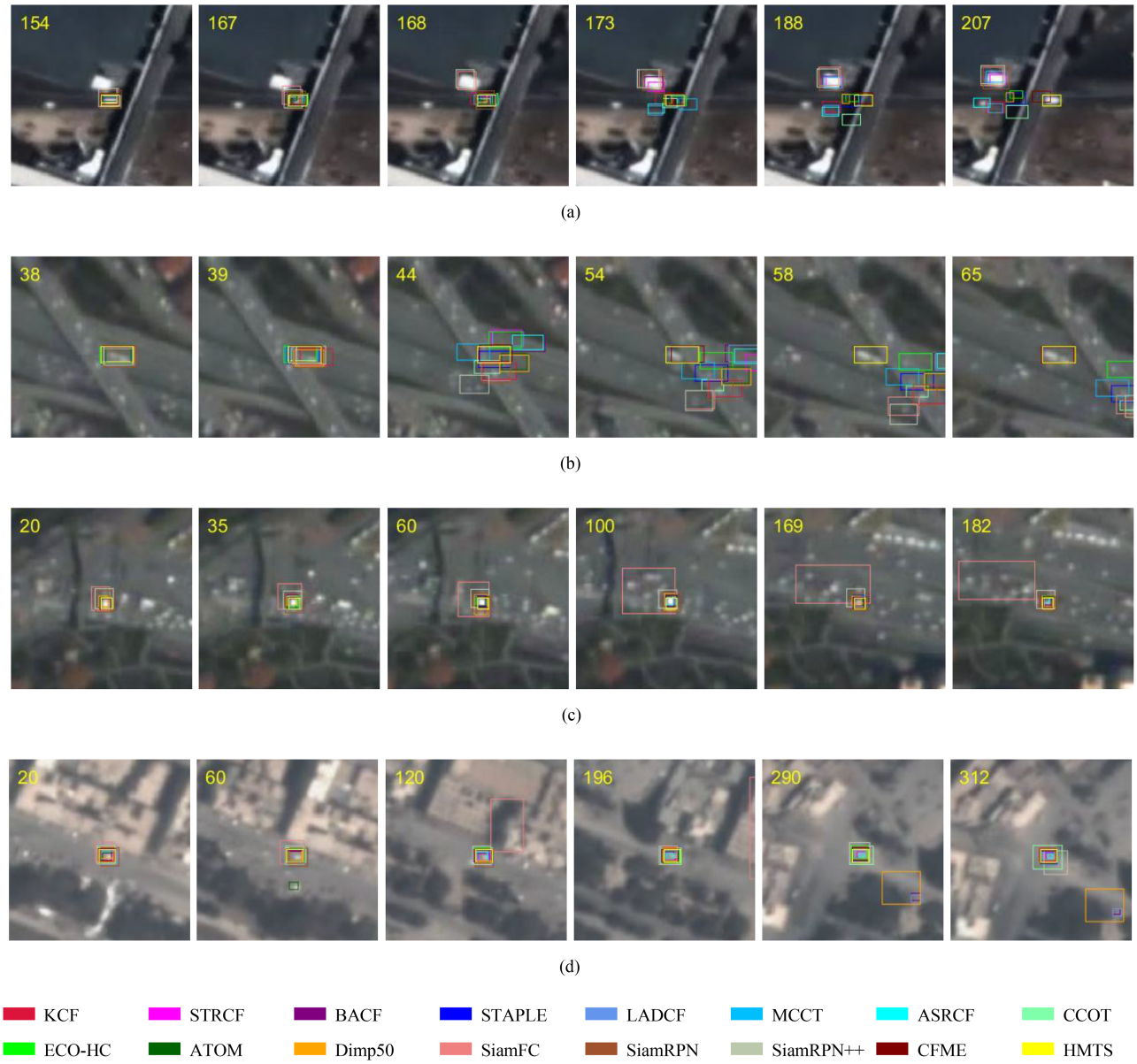



Fig. 4. Performance on challenging attributes. The number in the upper left corner is the frame number. The tracking boxes in different colors represent the tracking results of different trackers in the frame. (a) car_01. This video contains two challenges, ROT and FOC. (b) car_24. This video contains three challenges, ROT, POC, and FOC. (c) car_08. This video contains three challenges, BC, SOB, and TO. (d) car_26. This video contains three challenges, IV, BJT, and TO.

Success rate is an indicator used to evaluate the overlap between the tracking box and the ground truth. Suppose that the tracking box is $R_t$ and the ground truth is $R_0$, then the overlap of tracker is expressed as $S = |R_t \cap R_0|/|R_t \cup R_0|$ , where $|R_t \cap R_0|$ refers to the number of pixels at the intersection of $R_t$ and $R_0$, and $|R_t \cup R_0|$ refers to the number of pixels at the union of $R_t$ and $R_0$. The success rate is expressed as the percentage of frames whose overlap exceeds the overlap threshold. Also, the success plot is created. The horizontal axis of the success plot represents the overlap threshold, and the vertical axis indicates the success rate.

Based on this evaluation metric, the location error threshold is set to 5 pixels to sort the algorithms on precision. As for success rate, the average overlap score (AOE) is introduced as the basis for sorting, which is an average of overlap. Hence, all algorithms are evaluated from two perspectives.

*3) Implementation Details:* Our experiments are performed on a PC with 2.40-GHz CPU, and our tracker is implemented in MATLAB 2018a. The sample patch is 3.5 times the size of the object's ground truth. Besides, the standard deviation of the Gaussian surface, which is used to fit the regression targets, is 0.125. By comparison, the standard deviation of the kernel function is 0.6. The penalty parameter $\lambda$ is 1e-4, whereas the learning rates $\beta$ and $\gamma$ are both 0.02.

The features extracted in the tracker are grayscale feature and CN [52]. Additionally, the principal component analysis is conducted to simplify the features to ensure the computational speed of the tracker. The features of the vehicle on satellite

TABLE I
NINE ATTRIBUTES FOR VEHICLE ON SATSOT [49]

| Attribute | Definition |
|---|---|
| BC | *background clutter:* the background has similar appearance as the target |
| IV | *illumination variation:* the illumination of the target region changes significantly |
| LQ | *low quality:* the image is low quality and the target is difficult to be distinguished |
| ROT | *rotation:* the target rotates in the video |
| POC | *partial occlusion:* the target is partially occluded in the video |
| FOC | *full occlusion:* the target is fully occluded in the video |
| TO | *tiny object:* at least one ground truth bounding box has less than 25 pixels |
| SOB | *similar object:* there are objects of similar shape or same type around the target |
| BJT | *background jitter:* background jitter brings by the shaking of satellite camera |

TABLE II
OPE RESULTS OF TRACKERS FOR SENSITIVITY ANALYSIS

| Parameter | $\vartheta$ | | | | | $\theta$ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Value | 0 | 10 | 20 | 30 | 40 | 40 | 50 | 60 | 70 | 80 |
| Precision | 0.7156 | **0.7248** | 0.7096 | 0.6876 | 0.6826 | 0.7229 | 0.7248 | 0.7236 | **0.7253** | 0.7241 |
| AOE | 0.4159 | **0.4338** | 0.4257 | 0.4138 | 0.4109 | 0.4326 | 0.4338 | 0.4330 | **0.4344** | 0.4334 |

The bold entities indicate the best result of the comparison methods.

videos are weak and the size of the object is tiny. Normally, the texture feature is not suitable for representing the vehicle, e.g., HOG and local binary patterns [53]. This is mainly because these features must be extracted from the minimal cells, e.g., 4 pixels. Whereas, the object is too tiny to be divided into cells, so that it may make the loss of information and be meaningless. As stated by Shao et al. [12], the texture feature is ineffective for object tracking on satellite videos. As a result, simple features, such as grayscale and CN, are much more reliable.

### B. Sensitivity Analysis and Varying Criteria Comparison

In this section, we first conduct a sensitivity analysis to determine the most effective usage of the HM evaluation scheme. Then, we conduct varying criteria comparative experiments to compare PK with other high-confidence criteria.

In the sensitivity analysis, we vary the start frame $\vartheta$ of the HM evaluation scheme and test the sensitivity of $\vartheta$. Besides, we test the sensitivity of $\theta$ in (17). When test $\vartheta$, $\theta$ and the other parameters in the tracker are fixed. When test $\theta$, $\vartheta$ is set with the optimal one received from the last test and the other parameters in the tracker are also fixed. Table II shows the OPE results of two tests. The tracker works best when $\vartheta = 10$ and $\theta = 70$.

In the varying criteria comparative experiments, we compare the PK with other high-confidence criteria, including the average peak-to-correlation energy (APCE) [54], the peak, and the kurtosis. We designed four trackers, and Table III lists the OPE results of them. HMTS_APCE uses APCE as the high-confidence criterion. HMTS_PEAK and HMTS_KURTOSIS employ the peak and the kurtosis of the response map as the criterion individually. They are compared with our tracker HMTS, which employs both the peak and the kurtosis of the response map as the criterion.

TABLE III
OPE RESULTS OF TRACKERS FOR VARYING CRITERIA COMPARISON

| Tracker Name | Precision | AOE |
|---|---|---|
| HMTS_PEAK | 0.7084 | 0.4191 |
| HMTS_KURTOSIS | 0.7147 | 0.4285 |
| HMTS_APCE | 0.7026 | 0.4215 |
| HMTS (Ours) | **0.7253** | **0.4344** |

The bold entities indicate the best result of the comparison methods.

From the OPE results, it can be found that high-confidence criterion can indeed improve the precision and robustness, and the PK is more suitable for our application.

### C. Ablation Experiments

As mentioned earlier, our tracker treats KCF as the baseline, with the HM used to improve the update strategy of KCF. At the same time, KF is introduced for motion estimation, and the PK of the response map is used to detect whether the tracker drifts. To verify whether the HM, KF, and PK are effective in improving tracker performance, ablation experiments were conducted.

We design four trackers. Table IV lists the composition of them. Among these trackers, KCF_CN uses only CF, HMTS_NKFPK does not rely on KF and PK to correct the trajectory, and HMTS_NHM does not require HM for updating. They are compared with our tracker HMTS. The OPE results of these four groups of trackers are shown in columns 5 and 6 of Table IV.

From the performance of HMTS_NHM, it can be found out that the introduction of KF and PK improves the precision and

TABLE IV
COMPOSITIONS AND OPE RESULTS OF TRACKERS FOR ABLATION EXPERIMENTS

| Tracker Name | KCF | HM | KF + PK | Precision | AOE |
|---|---|---|---|---|---|
| KCF_CN | √ | | | 0.6753 | 0.4013 |
| HMTS_NKFPK | √ | √ | | 0.6693 | 0.3973 |
| HMTS_NHM | √ | | √ | 0.6895 | 0.4084 |
| HMTS (Ours) | √ | √ | √ | **0.7253** | **0.4344** |

The bold entities indicate the best result of the comparison methods.

success rate of the algorithm, which is because KF and PK terminate the tracker updating and predict the trajectory of the object when the tracker drifts.

From the performance of HMTS_NKFPK, it can be discovered that using HM plays no role in improving the precision and success rate, which is mainly because the baseline tracker tends to drift when the object is occluded and similar objects appear. Since the antidrift scheme is not used, the tracker does not stop updating. At the same time, the HM evaluation scheme scores HMs based on the drifting frame, and these HMs render new filter unreliable. Consequently, the tracker is less likely to return to normal, which reduces the performance of HMTS_NKFPK without KF and PK control.

Our tracker HMTS combines HM, KF, and PK. Also, the HM evaluation scheme keeps scoring HMs based on the correct sample. In this way, the updated filter will become more reliable. Hence, our tracker can take full advantage of HM.

In summary, the introduction of HM, KF, and PK is verified as effective. Our tracker shows a significant improvement over the baseline on the precision and success rate.

### D. Quantitative Comparison

Our tracker HMTS is evaluated with 16 state-of-the-art trackers, including CF-based tracker KCF [29], STRCF [40], BACF [41], STAPLE [55], LADCF [56], MCCT [57], ASRCF [58], CCOT [34], ECO-HC [35], ATOM [59], and DiMP [60]. And deep Siamese networks based tracker, SiamFC [46], SiamRPN [61], and SiamRPN++ [62]. And transformer-based tracker STARK [63]. Besides, an open-source satellite video tracker CFME [16] is also used for comparison with our tracker.

*1) Overall Results:* The overall OPE results are shown in Fig. 3. Our tracker HMTS ranks top on both indicators. The precision of HMTS is 0.7253, and the AOE of HMTS is 0.4344. Compared with the classical CF-based tracker KCF, our tracker HMTS improves the precision and AOE by 0.05 and 0.0331, respectively.

Meanwhile, as shown in the ranking list, CF-based trackers, e.g., KCF, MCCT, ECO-HC, STAPLE, and CFME, are advantageous over end-to-end networks based tracker, e.g., SiamFC and transformer-based tracker, e.g., STARK. That is to say, CF-based trackers are more suitable for vehicle tracking on satellite videos.

*2) Attributes-Based Results:* Overall, HMTS has excellent performance on most of challenges. Since the antidrift tracker correction scheme are effective on solving the object occlusion and similar objects appearance, it is no surprise that our tracker came out on top on POC, FOC, and SOB (see Tables V and VI). When these challenges arise, the antidrift tracker correction scheme plays a dominant role in preventing drift. When these challenges are addressed, the HM supports the tracker in returning to normal rapidly and precisely.

Meanwhile, tiny object is the object with a small size. There are few features on itself, so it is better to use background information as negative sample. The HMs make HMTS collect rich background information, which is favorable to tracking tiny object. Hence, our tracker has an excellent performance on TO (see Tables V and VI).

TABLE V
PARTIAL PRECISION ON CHALLENGING ATTRIBUTES

| Trackers | POC | FOC | TO | SOB |
|---|---|---|---|---|
| ASRCF | 0.5724 | 0.5409 | **0.6561** | 0.6129 |
| ATOM | 0.5417 | 0.5516 | 0.3983 | 0.5560 |
| BACF | 0.4863 | 0.5274 | 0.4835 | 0.4641 |
| CCOT | 0.5805 | **0.6133** | **0.6979** | 0.6743 |
| CFME | **0.6883** | **0.7495** | 0.6628 | **0.7327** |
| Dimp50 | 0.5418 | 0.5681 | 0.5541 | 0.5864 |
| ECO-HC | 0.5923 | 0.5491 | 0.6072 | 0.6625 |
| KCF | 0.6021 | 0.5828 | 0.6234 | 0.6452 |
| LADCF | 0.6028 | 0.5767 | 0.6215 | 0.7014 |
| MCCT | 0.5801 | 0.5457 | 0.6421 | **0.7161** |
| SiamFC | 0.4767 | 0.5100 | 0.2970 | 0.4295 |
| SiamRPN | 0.4975 | 0.5579 | 0.4660 | 0.5954 |
| SiamRPN++ | 0.5116 | 0.5201 | 0.4825 | 0.6323 |
| STAPLE | 0.5692 | 0.5453 | 0.6117 | 0.6864 |
| STARK | 0.3848 | 0.5017 | 0.2060 | 0.3217 |
| STRCF | **0.6194** | 0.5463 | 0.6333 | 0.7120 |
| **HMTS (ours)** | **0.7147** | **0.7708** | **0.6901** | **0.7918** |

The bold entities indicate the best result of the comparison methods.

TABLE VI
PARTIAL SUCCESS RATE ON CHALLENGING ATTRIBUTES

| Trackers | POC | FOC | TO | SOB |
|---|---|---|---|---|
| ASRCF | 0.3268 | 0.2495 | 0.2819 | 0.3293 |
| ATOM | 0.3465 | 0.2904 | 0.2229 | 0.3510 |
| BACF | 0.2945 | 0.2624 | 0.2392 | 0.2725 |
| CCOT | 0.3482 | 0.2975 | 0.3253 | 0.3672 |
| CFME | **0.4301** | **0.4327** | **0.3664** | **0.4393** |
| Dimp50 | 0.3594 | 0.3049 | 0.2986 | 0.3579 |
| ECO-HC | 0.3825 | 0.2894 | 0.3316 | 0.3650 |
| KCF | 0.3797 | **0.3219** | 0.3280 | 0.3694 |
| LADCF | 0.3490 | 0.2893 | 0.3169 | 0.3647 |
| MCCT | 0.3735 | 0.2889 | **0.3581** | **0.4296** |
| SiamFC | 0.2687 | 0.2158 | 0.0936 | 0.2043 |
| SiamRPN | 0.2875 | 0.2522 | 0.1716 | 0.3184 |
| SiamRPN++ | 0.2948 | 0.2400 | 0.1739 | 0.3286 |
| STAPLE | 0.3694 | 0.2856 | 0.3367 | 0.4064 |
| STARK | 0.2452 | 0.2503 | 0.0732 | 0.1880 |
| STRCF | **0.3833** | 0.2881 | 0.3229 | 0.3944 |
| **HMTS (ours)** | **0.4488** | **0.4453** | **0.3677** | **0.4514** |

The bold entities indicate the best result of the comparison methods.

### E. Qualitative Comparison

In order to better demonstrate the performance of each tracker, qualitative experiments are carried out, with the tracking box of each tracker at each frame displays the video frame. Partial performance is shown in Fig. 4.

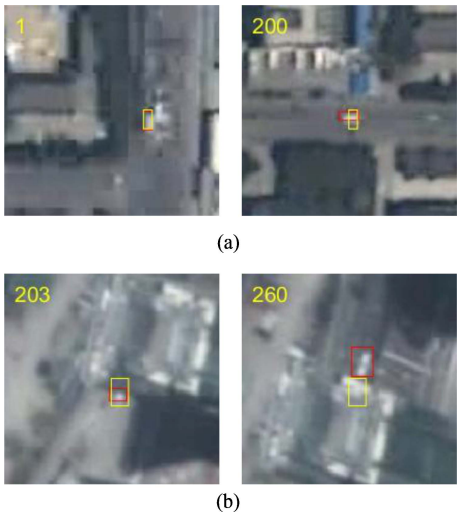



Fig. 5. Inferior cases. The number in the upper left corner is the frame number. The yellow box is tracking box and the red box is ground truth. (a) Size change. (b) Motion change

*1) Qualitative Analysis:* From video car_01 and car_24, it can be seen that HMTS is effective in solving the occlusion issue while most of trackers drift when hitting the obstacle. In video car_01, the object is occluded at frame 173, only CFME and HMTS overcome this challenge. The reason is CFME and HMTS use the motion estimation scheme, whereas other trackers do not perceive the disappearance of the object and collect false information when object is occluded. Meanwhile, the time of occlusion is long, and the object moves a long distance. Hence, when the object occurs again, the object is not in the ROI of them, and it is difficult for them to detect the object after collecting so much false information. On the contrary, the antidrift tracker correction scheme makes HMTS stop collecting information and predict the object's position when object is being occluded. Even with long occlusions, the object will be in our ROI. This scheme not only makes HMTS reliable but also leads to a continuous trajectory.

Meanwhile, from video car_08 and car_26, it can be found out that the performance of our tracker is superior in case of SOB, TO, IV, etc. Although the feature of object itself is insufficient to identify it, HM is beneficial for HMTS to collect the background information. As a result, HMTS can ensure accurate tracking.

*2) Inferior Cases:* Although HMTS is verified as the optimal tracker, there are some inferior cases within the tracking process (see Fig. 5).

HMTS does not use the scale-adaptive mechanism, which leads to the tracking box cannot be adjusted along with the change of the object size, as shown in Fig. 5(a). It results in a lower success rate. Besides, the prediction function of KF is limited. If the velocity and orientation of the object change during occlusion, the KF prediction is likely to be unreliable and may result in drift, as illustrated in Fig. 5(b).

## V. Conclusion

In this article, a tracker (HMTS) is proposed for vehicle tracking on satellite videos. In addition, an HM evaluation scheme is proposed for the evaluation of HMs by using cross-correlation function. Besides, to prevent tracker drift, an antidrift tracker correction scheme is proposed. Our tracker is compared with 16 state-of-the-art trackers. As demonstrated by the quantitative and qualitative experiments, HMTS produces an excellent performance. Additionally, sensitivity analysis, varying criteria comparative experiments, and ablation experiments reveal that HM, KF, and PK are effective in improving the accuracy and robustness of the algorithm.

## References

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

[2] 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, doi: 10.1109/TPAMI.2014.2388226.

[3] M. Kristan et al., "The ninth visual object tracking VOT2021 challenge results," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops*, Montreal, BC, Canada, 2021, pp. 2711–2738.

[4] C. Deng, S. He, Y. Han, and B. Zhao, "Learning dynamic spatial-temporal regularization for UAV object tracking," *IEEE Signal Process. Lett.*, vol. 28, pp. 1230–1234, Jun. 2021, doi: 10.1109/LSP.2021.3086675.

[5] D. Yuan, X. Chang, Z. Li, and Z. He, "Learning adaptive spatial-temporal context-aware correlation filters for UAV tracking," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 18, no. 3, Aug. 2022, Art. no. 70, doi: 10.1145/3486678.

[6] Q. Liu, D. Yuan, N. Fan, P. Gao, X. Li, and Z. He, "Learning dual-level deep representation for thermal infrared tracking," *IEEE Trans. Multimedia*, to be published, doi: 10.1109/TMM.2022.3140929.

[7] B. Zhao, Y. Han, H. Wang, L. Tang, X. Liu, and T. Wang, "Robust shadow tracking for video SAR," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 5, pp. 821–825, May 2021, doi: 10.1109/LGRS.2020.2988165.

[8] L. Tang, W. Tang, X. Qu, Y. Han, W. Wang, and B. Zhao, "A scale-aware pyramid network for multi-scale object detection in SAR images," *Remote Sens.*, vol. 14, no. 4, Feb. 2022, Art. no. 973, doi: 10.3390/rs14040973.

[9] 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, doi: 10.11947/j.AGCS.2017.20160599.

[10] B. Du, S. Cai, and C. Wu, "Object tracking in satellite videos based on a multiframe optical flow tracker," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 8, pp. 3043–3055, Aug. 2019, doi: 10.1109/JSTARS.2019.2917703.

[11] B. Du, Y. Sun, S. Cai, C. Wu, and Q. Du, "Object tracking in satellite videos by fusing the kernel correlation filter and the three-frame-difference algorithm," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 2, pp. 168–172, Feb. 2018, doi: 10.1109/LGRS.2017.2776899.

[12] J. Shao, B. Du, C. Wu, and L. Zhang, "Can we track targets from space? A hybrid kernel correlation filter tracker for satellite video," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 11, pp. 8719–8731, Nov. 2019, doi: 10.1109/TGRS.2019.2922648.

[13] J. Shao, B. Du, C. Wu, and L. Zhang, "Tracking objects from satellite videos: A velocity feature based correlation filter," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 7860–7871, Oct. 2019, doi: 10.1109/TGRS.2019.2916953.

[14] Y. Guo, D. Yang, and Z. Chen, "Object tracking on satellite videos: A correlation filter-based tracking method with trajectory correction by Kalman filter," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 9, pp. 3538–3551, Sep. 2019, doi: 10.1109/JSTARS.2019.2933488.

[15] J. Wu et al., "Satellite video point-target tracking based on Hu correlation filter," *Chin. Space Sci. Technol.*, vol. 39, no. 3, pp. 55–63, Jun. 2019, doi: 10.16708/j.cnki.1000-758X.2018.0059.

[16] S. Xuan, S. Li, M. Han, X. Wan, and G. Xia, "Object tracking in satellite videos by improved correlation filters with motion estimations," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 2, pp. 1074–1086, Feb. 2020, doi: 10.1109/TGRS.2019.2943366.

[17] Y. Wang, T. Wang, G. Zhang, Q. Cheng, and J. Wu, "Small target tracking in satellite videos using background compensation," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 10, pp. 7010–7021, Oct. 2020, doi: 10.1109/TGRS.2020.2978512.

[18] J. Shao, B. Du, C. Wu, and Y. Pingkun, "PASiam: Predicting attention inspired Siamese network, for space-borne satellite video tracking," in *Proc. IEEE Int. Conf. Multimedia Expo.*, Shanghai, China, Jul. 2019, pp. 1504–1509.

[19] Z. Hu, D. Yang, K. Zhang, and Z. Chen, "Object tracking in satellite videos based on convolutional regression network with appearance and motion features," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 783–793, Feb. 2020, doi: 10.1109/JSTARS.2020.2971657.

[20] Y. Cui, B. Hou, Q. Wu, B. Ren, S. Wang, and L. Jiao, "Remote sensing object tracking with deep reinforcement learning under occlusion," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Jul. 2021, Art. no. 5605213, doi: 10.1109/TGRS.2021.3096809.

[21] W. Zhang, L. Jiao, F. Liu, L. Li, X. Liu, and J. Liu, "MBLT: Learning motion and background for vehicle tracking in satellite videos," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Sep. 2021, Art. no. 4703315, doi: 10.1109/TGRS.2021.3109028.

[22] C. Deng, D. Jing, Y. Han, S. Wang, and H. Wang, "FAR-Net: Fast anchor refining for arbitrary-oriented object detection," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, Jan. 2022, Art. no. 6505805, doi: 10.1109/LGRS.2022.3144513.

[23] K. Du, Y. Ju, Y. Jin, G. Li, Y. Li, and S. Qian, "Object tracking based on improved MeanShift and SIFT," in *Proc. Int. Conf. Consum. Electron. Commun. Netw.*, Yichang, China, 2012, pp. 2716–2719.

[24] D. Exner, E. Bruns, D. Kurz, A. Grundhöfer, and O. Bimber, "Fast and robust CAMShift tracking," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, San Francisco, CA, USA, 2010, pp. 9–16.

[25] K. Nummiaro, E. Koller-Meier, and L. Van Gool, "An adaptive color-based particle filter," *Image Vis. Comput.*, vol. 21, no. 1, pp. 99–110, Jan. 2003, doi: 10.1016/S0262-8856(02)00129-4.

[26] D. Sun, S. Roth, and M. J. Black, "Secrets of optical flow estimation and their principles," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, USA, 2010, pp. 2432–2439.

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

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

[29] 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, doi: 10.1109/TPAMI.2014.2345390.

[30] K. Zhang, L. Zhang, M. Yang, and D. Zhang, "Fast tracking via spatio-temporal context learning," Nov. 2013, *arXiv:1311.1939*.

[31] C. Deng, Y. Han, and B. Zhao, "High-performance visual tracking with extreme learning machine framework," *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2781–2792, Jun. 2020, doi: 10.1109/TCYB.2018.2886580.

[32] D. Yuan, X. Chang, P. Huang, Q. Liu, and Z. He, "Self-supervised deep correlation tracking," *IEEE Trans. Image Process.*, vol. 30, pp. 976–985, Dec. 2020, doi: 10.1109/TIP.2020.3037518.

[33] K. Yang et al., "SiamCorners: Siamese corner networks for visual tracking," *IEEE Trans. Multimedia*, vol. 24, pp. 1956–1967, Apr. 2021, doi: 10.1109/TMM.2021.3074239.

[34] 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.*, Amsterdam, The Netherlands, 2016, pp. 472–488.

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

[36] 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.*, Columbus, OH, USA, 2014, pp. 1090–1097.

[37] Y. Han, C. Deng, Z. Zhang, J. Li, and B. Zhao, "Adaptive feature representation for visual tracking," in *Proc. IEEE Int. Conf. Image Process.*, Beijing, China, 2017, pp. 1867–1870.

[38] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. Brit. Mach. Vis. Conf.*, Nottingham, U.K., 2014, pp. 1–5.

[39] Z. Zhao, Y. Han, T. Xu, X. Li, H. Song, and J. Luo, "A reliable and real-time tracking method with color distribution," *Sensors*, vol. 17, no. 10, Oct. 2017, Art. no. 2303, doi: 10.3390/s17102303.

[40] F. Li, C. Tian, W. Zuo, L. Zhang, and M.-H. Yang, "Learning spatial-temporal regularized correlation filters for visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, 2018, pp. 4904–4913.

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

[42] Y. Han, C. Deng, B. Zhao, and B. Zhao, "Spatial-temporal context-aware tracking," *IEEE Signal Process. Lett.*, vol. 26, no. 3, pp. 500–504, Mar. 2019, doi: 10.1109/LSP.2019.2895962.

[43] Y. Li, C. Fu, F. Ding, Z. Huang, and G. Lu, "AutoTrack: Towards high-performance visual tracking for UAV with automatic spatio-temporal regularization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Seattle, WA, USA, 2020, pp. 11920–11929.

[44] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Sep. 2015, *arXiv:1409.1556*.

[45] 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. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, 2017, pp. 5000–5008.

[46] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional Siamese networks for object tracking," in *Proc. Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands, 2016, pp. 850–865.

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

[48] Q. Yin et al., "Detecting and tracking small and dense moving objects in satellite videos: A benchmark," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Nov. 2021, Art. no. 5612518, doi: 10.1109/TGRS.2021.3130436.

[49] M. Zhao, S. Li, S. Xuan, L. Kou, S. Gong, and Z. Zhou, "SatSOT: A benchmark dataset for satellite video single object tracking," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Jan. 2022, Art. no. 5617611, doi: 10.1109/TGRS.2022.3140809.

[50] Y. Han, C. Deng, B. Zhao, and D. Tao, "State-aware anti-drift object tracking," *IEEE Trans. Image Process.*, vol. 28, no. 8, pp. 4075–4086, Aug. 2019, doi: 10.1109/TIP.2019.2905984.

[51] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 564–577, May 2003, doi: 10.1109/TPAMI.2003.1195991.

[52] J. van de Weijer, C. Schmid, J. Verbeek, and D. Larlus, "Learning color names for real-world applications," *IEEE Trans. Image Process.*, vol. 18, no. 7, pp. 1512–1523, Jul. 2009, doi: 10.1109/TIP.2009.2019809.

[53] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002, doi: 10.1109/TPAMI.2002.1017623.

[54] M. Wang, Y. Liu, and Z. Huang, "Large margin object tracking with circulant feature maps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, 2017, pp. 4800–4480.

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

[56] T. Xu, Z. Feng, X. Wu, and J. Kittler, "Learning adaptive discriminative correlation filters via temporal consistency preserving spatial feature selection for robust visual object tracking," *IEEE Trans. Image Process.*, vol. 28, no. 11, pp. 5596–5609, Nov. 2019, doi: 10.1109/TIP.2019.2919201.

[57] N. Wang, W. Zhou, Q. Tian, R. Hong, M. Wang, and H. Li, "Multi-cue correlation filters for robust visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, 2018, pp. 4844–4853.

[58] K. Dai, D. Wang, H. Lu, C. Sun, and J. Li, "Visual tracking via adaptive spatially-regularized correlation filters," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, 2019, pp. 4665–4674.

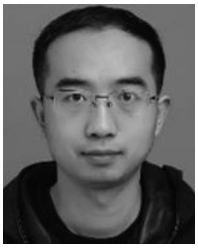
[59] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ATOM: Accurate tracking by overlap maximization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, 2019, pp. 4655–4664.

[60] G. Bhat, M. Danelljan, L. Van Gool, and R. Timofte, "Learning discriminative model prediction for tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Seoul, South Korea, 2019, pp. 6181–6190.

[61] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with Siamese region proposal network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, 2018, pp. 8971–8980.
[62] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "SiamRPN++: Evolution of Siamese visual tracking with very deep networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, 2019, pp. 4277–4286.
[63] B. Yan, H. Peng, J. Fu, D. Wang, and H. Lu, "Learning spatio-temporal transformer for visual tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Montreal, QC, Canada, 2021, pp. 10428–10437.

**Shili Chen** received the B.E. degree in remote sensing and information engineering in 2021 from Wuhan University, Wuhan, China, where she is currently working toward the M.E. degree in photogrammetry and remote sensing with the School of Remote Sensing and Information Engineering.

Her research interests include object tracking and machine learning.

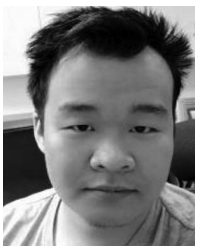


**Taoyang Wang** received the B.E. and Ph.D.E. degrees in photogrammetry and remote sensing from the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, China, in 2007 and 2012, respectively. His doctoral dissertation concerned the block adjustment of high-resolution satellite remote sensing imagery.

He has been with the School of Remote Sensing and Information Engineering, Wuhan University, since 2014, where he became an Associate Research Fellow in 2015. His research interests include space photogrammetry, geometry processing of spaceborne optical/SAR/InSAR imagery, target detection, and recognition based on satellite video.

**Hongshuo Wang** received the M.E. degree in electronic and communication engineering from the School of Information and Electronics, Beijing Institute of Technology, Beijing, China, in 2021.

She is currently with the Beijing Aerospace Automatic Control Institute, Beijing, China. Her research interests include video target detection and tracking.



**Yunming Wang** received the B.E. degree in surveying and mapping from Henan Polytechnic University, Jiaozuo, China, in 2017, and the M.E. degree in surveying and mapping in 2020 from Wuhan University, Wuhan, China where he is currently working toward the Ph.D.E. degree with the School of Remote Sensing and Information Engineering.

His research interest mainly focuses on 3-D reconstruction.

**Jianzhi Hong** received the B.S. degree in geographic information science from the School of Geography, South China Normal University, Guangzhou, China, in 2020, and the M.E. degree in resources and environment in 2022 from the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, China, where he is currently working toward the Doctorate degree in photogrammetry and remote sensing.

His main research direction is intelligent interpretation of moving objects in optical satellite video, including detection, tracking, and segmentation.

**Tiancheng Dong** received the B.S. degree in remote sensing science and technology and the M.S. degree in geophysical prospecting and information technology from the China University of Geosciences, Wuhan, China, in 2021. He is currently working toward the Ph.D. degree in photogrammetry and remote sensing from the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan, China.

His research mainly focuses on ship detection of SAR image and preprocessing of Video-SAR.

**Zhen Li** received the B.S. degree in remote sensing science and technology and the M.S. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2009 and 2011, respectively.

She is currently an Optical Satellite System Senior Engineer with the Institute of Remote Sensing Satellite, China Academy of Space Technology, Beijing, China. Her research interests include satellite mission analysis, high geometric accuracy system design, and hyperspectral satellite system design and applications.