remote sensing

MDPI

Article

Adaptive and Anti-Drift Motion Constraints for Object Tracking in Satellite Videos

Junyu Fan and Shunping Ji *

School of Remote Sensing and Information Engineering, Wuhan University, 129 Luoyu Road, Wuhan 430079, China; fanjunyu@whu.edu.cn
* Correspondence: jishunping@whu.edu.cn

Abstract: Object tracking in satellite videos has garnered significant attention due to its increasing importance. However, several challenging attributes, such as the presence of tiny objects, occlusions, similar objects, and background clutter interference, make it a difficult task. Many recent tracking algorithms have been developed to tackle these challenges in tracking a single interested object, but they still have some limitations in addressing them effectively. This paper introduces a novel correlation filter-based tracker, which uniquely integrates attention-enhanced bounding box regression and motion constraints for improved single-object tracking performance. Initially, we address the regression-related interference issue by implementing a spatial and channel dual-attention mechanism within the search area's region of interest. This enhancement not only boosts the network's perception of the target but also improves corner localization. Furthermore, recognizing the limitations in small size and low resolution of target appearance features in satellite videos, we integrate motion features into our model. A long short-term memory (LSTM) network is utilized to create a motion model that can adaptively learn and predict the target's future trajectory based on its historical movement patterns. To further refine tracking accuracy, especially in complex environments, an anti-drift module incorporating motion constraints is introduced. This module significantly boosts the tracker's robustness. Experimental evaluations on the SatSOT and SatVideoDT datasets demonstrate that our proposed tracker exhibits significant advantages in satellite video scenes compared to other recent trackers for common scenes or satellite scenes.

Keywords: single object tracking; satellite video; correlation filter; motion constraints

Citation: Fan, J.; Ji, S. Adaptive and Anti-Drift Motion Constraints for Object Tracking in Satellite Videos. *Remote Sens.* **2024**, *16*, 1347. https://doi.org/10.3390/rs16081347

Academic Editor: Gong Cheng

Received: 20 February 2024
Revised: 6 April 2024
Accepted: 8 April 2024
Published: 11 April 2024

1. Introduction

Visual object tracking, predicting the dynamic state of targets based on initial video frame cues, is fundamental research for applications such as visual surveillance [1], human-computer interaction [2], and autonomous driving [3,4]. In the Earth observation field, remarkable advancements in video satellite technology [5,6] have been witnessed in recent years. By employing a stare observation approach [7], video satellites are capable of continuously observing specific regions, providing valuable video data of the Earth's surface. The development has facilitated the emergence of a new task: tracking an interested object using satellite videos. This task enables real-time monitoring and tracking of various objects of interest, such as vehicles, aircraft, ships, and trains, on a broad region of the Earth's surface, leading to a wide range of substantial applications including traffic and environmental monitoring [8], military reconnaissance [9], and disaster management [10].

However, tracking objects in satellite videos presents a series of complex challenges. Satellite video frames typically offer a wide field of view, capturing small, low-resolution objects. Consequently, this results in a pervasive lack of texture, color, and other distinguishing features that are essential for accurate target perception. Furthermore, single object tracking in satellite videos frequently encounters issues of occlusion and interference. While these issues also exist in general tracking scenarios, they are intensified in the

satellite context due to the top-down viewing angle. For instance, moving vehicles often face full occlusion from buildings or natural landscapes in satellite videos. Moreover, interference factors, such as atmospheric turbulence, changes in illumination, similar objects, and background clutter, further hinder the object tracking process, leading to tracking drift.

Single object tracking algorithms are broadly classified into three categories: correlation filter-based [11,12], Siamese network-based [13,14], and Transformer-based [15,16]. Correlation filter-based methods optimize target models using historical frames, constructing efficient filters that precisely localize targets by correlating them with the search region in the current frame. Siamese network-based approaches view object tracking as a similarity learning problem, focusing on offline training of shared feature networks used to extract similarity features between the target template and the search region. The more recent Transformer-based trackers utilize attention mechanisms for global interaction of features between the template and the search region, resulting in a significant representation of the target. These algorithms typically rely on robust visual feature descriptions for efficacy.

This paper aims to enhance object tracking performance in challenging satellite video scenes. A significant limitation that exists in current tracking algorithms based on Siamese network and Transformer is their lack of inherent mechanisms for online target template updating. This shortfall is particularly problematic in satellite video contexts, where the appearance features of objects frequently change due to interference factors, making the tracking task more complex and demanding. In response to this issue, our research turns to trackers based on correlation filters. These trackers are inherently equipped with the ability for online optimization of the target model, a capability that allows them to adeptly adapt to changing target appearances. This attribute is a substantial advantage in the context of satellite video tracking. We develop our tracker based on the correlation filter framework, specifically building it upon the DiMP [17]. The baseline tracker regresses target bounding boxes based on dense sampling proposals strategy. However, the dense sampling process in satellite video scenes potentially introduces a significant number of false samples due to small-size targets and various interferences. These false samples can greatly hinder the accurate prediction of optimal bounding box proposals, leading to tracking drift. In this study, we firstly improve the bounding box regression strategy. Subsequently, we implement a motion model based on a long short-term memory network (LSTM) [18] and integrate an anti-drift strategy, aiming to enhance target position prediction and effectively reduce tracking drift. In summary, our contributions include:

1. We propose an attention-enhanced bounding box regression branch to improve the baseline tracker's performance in satellite video scenes. Instead of dense sampling proposals, we focus on the regions of interest in the search area and integrate a dual spatial and channel attention mechanism to enhance the tracker's perception ability of the target.
2. We design a learnable motion model based on LSTM to realize trajectory distribution estimation. The model effectively utilizes the historical trajectory of the target to extract short-term motion features for estimating the trajectory as well as the motion trend distribution. It serves to compensate for the common issue of limited appearance features in satellite video scenes.
3. We propose an anti-drift module for satellite video single object tracking, which models the difference between the observation distribution and motion trend distribution of the target to detect drift. By incorporating this module, we appropriately introduce motion constraints during the tracking process, effectively solving the drift problem and improving overall tracking performance.

2. Related Literature

2.1. Correlation Filter-Based Object Tracking

The correlation filter-based trackers are fundamentally centered on predicting the optimal parameters of a filter model from target samples through online optimization. The highest response of cross-correlation on the search region of the current frame with

the template region indicates the target's location. MOSSE [11] is a pioneering work that introduced correlation filters to the field of object tracking. To enhance the robustness of models, MOSSE employs sparse sampling to generate multiple training samples. CSK [19] implicitly constructs a dense sampling matrix by cyclically shifting to replace MOSSE's sparse sampling strategy. The approach increases the number of training samples while reducing redundancy. KCF [12], building upon CSK, introduces multi-channel HoG [20] features as a replacement for grayscale features and employs a kernel function, resulting in enhanced tracker performance. SAMF [21] combines features from HoG, color names [22], and grayscale, while employing a scaling pool strategy to perform cross-correlation on multi-scale candidate regions to enable adaptive target scale estimation. Recognizing that cyclically shifted samples may produce negative samples with boundary effects that do not accurately represent real-world negative samples, BACF [23] improves sample quality by sampling true negative samples from the background to optimize the prediction model. C-COT [24] integrates the deep neural network VGG [25] for feature extraction. To address the issue of different resolutions in feature maps from various convolutional layers, C-COT employs frequency domain implicit interpolation to extend feature maps of different resolutions to a continuous spatial domain while maintaining high localization accuracy. ECO [26] employs a factorized convolution approach for simplified feature extraction. Additionally, ECO uses a Gaussian mixture model for grouping and simplifying the training sample set, ensuring diversity while effectively avoiding overfitting of the filter model.

With the evolution of deep learning, some of the latest correlation filter algorithms have begun adopting offline end-to-end training modes to enhance algorithm performance. CFNet [27] combines convolutional neural networks with correlation filters, enabling end-to-end training of correlation filter-based trackers. DiMP, PrDiMP [28], and SuperDiMP [29] apply the end-to-end training strategy to the filter-based prediction module. They effectively utilize large-scale datasets to perform offline training on the prediction module, thereby enhancing the model's discriminative capability of the prediction model for targets and backgrounds.

2.2. Siamese Network-Based Object Tracking

Siamese convolutional neural network trackers represent a distinct paradigm from the correlation filter trackers. SiamFC [13] pioneered the use of an end-to-end offline-trained fully convolutional Siamese network for object tracking. It views tracking as a similarity learning problem and, through extensive offline training, transforms the Siamese network into a universal similarity evaluator. During online tracking, it estimates the similarity between the template and the search region features using the Siamese network, thereby determining the target's position and obtaining the optimal bounding box through a multi-scale strategy. SiamRPN [30] incorporates the region proposal network (RPN) [31] structure from object detection into the Siamese network tracking framework. In SiamRPN++ [14], the spatial distribution of targets is adjusted by uniform sampling from training samples, reducing bias towards positive samples being centered in the search region. It incorporates multi-level feature aggregation, with shallow layers for localization and deep layers for semantic encoding. DaSiamRPN [32], focusing on the balance between positive and negative samples during training, enhances the tracker's discriminative capability and generalization performance by increasing positive samples and hard negative samples. Both SiamCAR [33] and SiamBAN [34] introduce anchor-free strategies to replace the RPN structure, reducing the number of hyperparameters and simplifying parameter settings during training. CGACD [35] utilizes correlation-guided attention during the regression phase and develops a high-performance corner-based bounding box regression method.

2.3. Transformer-Based Object Tracking

With the effective feature representation facilitated by attention mechanisms, Transformer [36] has gained significant recognition in the field of computer vision, and Transformer-

based object tracking algorithms have attracted substantial attention from researchers. Transt [37] introduces an attention-based feature fusion network, which employs both self-attention and cross-attention to combine template and search region features, replacing correlation operations, and resulting in a more effective representation of the similarity. STARK [15] leverages Transformer to combine spatial and temporal features of the target, generating global spatio-temporal feature dependencies for target localization. Additionally, STARK simplifies the tracking pipeline by eliminating post-processing steps such as cosine window and bounding box smoothing. TCTrack [38] proposes an online temporally adaptive convolution in the feature extraction stage to incorporate temporal information for enhancing spatial features. In the refinement of similarity maps, an adaptive temporal Transformer is introduced to effectively encode and decode temporal context information between consecutive frames. Mixformer [16] employs the Mix-Attention Module to simultaneously perform template and search region feature extraction and information interaction. Additionally, it utilizes a custom asymmetric attention strategy to eliminate unnecessary cross-attention regions while ensuring template robustness by implementing a score-based template updating mechanism.

2.4. Single Object Tracking in Satellite Video

Due to the distinctive characteristics of satellite videos as compared to common videos, object tracking algorithms developed for common videos often yield unsatisfactory results when directly applied to satellite videos. Researchers have explored various tracking algorithms for satellite videos.

Du et al. [39] propose a tracker for satellite video that combines kernelized correlation filters (KCF) with a three-frame difference motion detection method. KCF locates the target based on appearance features, while the three-frame difference method introduces target motion features into the tracking process. Liu et al. [40] use a second-order parabolic model based on Taylor expansion to fit discrete target response values, achieving sub-pixel precision in target localization. It introduces an adaptive Kalman filter (AKF) to compensate and correct tracking results. Du et al. [41] employ a multi-frame differential optical flow method to distinguish moving objects from the background. It utilizes the HSV color system to describe motion targets based on the optical flow field and employs integral image techniques to precisely locate the tracking targets. Xuan et al. [42] combine Kalman filtering and motion trajectory averaging methods, introducing prior target state information into the KCF tracking process, which effectively reduces performance degradation caused by boundary effects and alleviates the problem of tracking failure caused by occlusion. Li et al. [43] introduce feature fusion (HOG, color names, and color histograms) to boost the performance of the correlation filter-based tracker in complex scenes. It also presents a cost-effective motion estimation method, combining Kalman and particle filters, to enhance tracker robustness against occlusion in linear and nonlinear motion scenes. Shao et al. [44] introduce a lightweight parallel high-resolution network to detect small targets in satellite videos, enhancing tracking precision. Additionally, a pixel-level refinement model based on motion target detection and adaptive fusion is proposed to improve tracking robustness. Hu et al. [45] construct a model regression network using a single-layer convolutional structure. It extracts target appearance features from the VGG network and fuses them with optical flow motion features. The network is updated online using gradient descent, leveraging both background information and motion features for target tracking. Yang et al. [46] adaptively fuse multi-stage feature cross-correlation response maps to enhance localization for small targets. Chen et al. [47] fuse response maps for target texture and spectral features to enhance discrimination between the target and the background.

In the current single object tracking methods for satellite videos, strategies like feature fusion, super-resolution, and response fusion are frequently employed to adapt to satellite video scenes, primarily aiming to enhance foreground–background discrimination. The impact of the bounding box regression process on tracker performance has not been sufficiently considered. In this context, drawing inspiration from the CGACD [35], this

paper introduces satellite scene-specific enhancements to the DiMP regression process. Moreover, the challenges in satellite videos can directly or indirectly lead to tracking drift. Existing research typically introduces target motion information into the tracking process using motion models [40,42,43,47] or motion detection methods [39,41,44,45]. These methods typically rely on specific motion assumptions, restricting the model's generalization performance. We introduce a trajectory distribution estimation strategy to model target motion and learn motion patterns adaptively. Simultaneously, we present a learnable anti-drift strategy in conjunction with the trajectory distribution estimation strategy, to incorporate motion constraints into the tracking process, ensuring robustness across different scenes.

3. Methodology

In this section, we present the details of our proposed tracking method for satellite video scenes, comprising three main components: the target classification branch (TCB), the attention-enhanced regression branch (AERB), and the motion constraint branch (MCB), which consists of the trajectory distribution estimation module (TDEM) and the anti-drift module (ADM).

As illustrated in Figure 1, TCB leverages features of the training set to generate a discriminative target model, applying this model to classify the search region in the test frame and thereby determine the target observation center. Based on the target observation center and associated features, AERB regresses the target's bounding box in the test frame. MCB utilizes TDEM to extract trajectory motion features from the historical states of previous n frames for estimating the target's trajectory as well as the motion trend distribution, which are then applied by ADM to constrain and compensate for the result of regression, enhancing the overall accuracy and robustness of the tracking process.

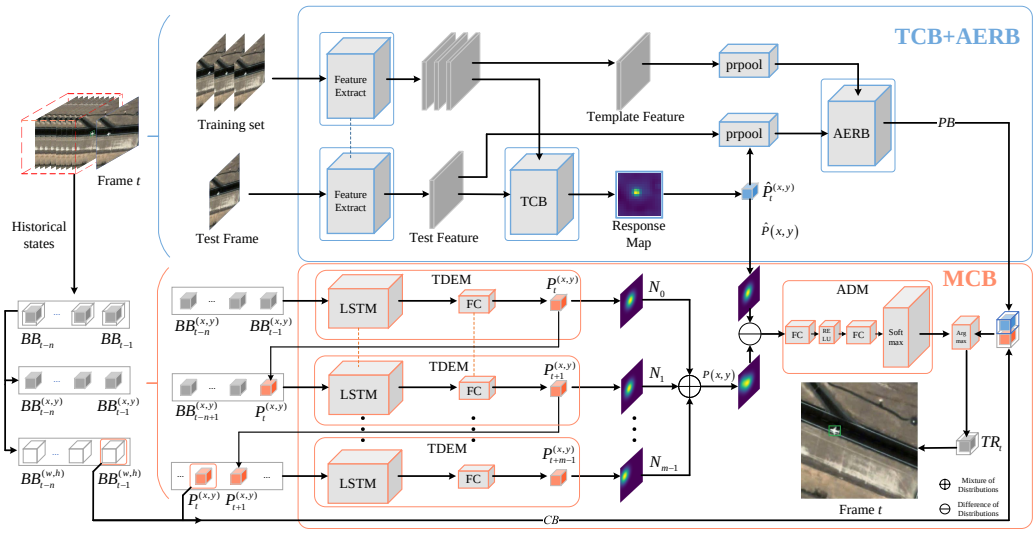

Figure 1. The architecture of the proposed tracker. TCB takes the output of the feature extraction network as input, and combines AERB to observe the target state in the current frame. In MCB, TDEM uses the historical motion information of the target to estimate the trajectory distribution, ADM detects the tracking state and introduces motion constraints into the tracking process according to the trajectory distribution estimated by TDEM.

3.1. Target Classification Branch

The baseline method DiMP consists of two branches: the target classification branch and the bounding box regression branch. The target classification branch (TCB) is mainly employed to predict a discriminative target model f, which is embedded in the form of the convolutional kernel within a convolutional layer and classifies the foreground and background in the test frame. To achieve online prediction of the target model, while

ensuring computational efficiency, the TCB utilizes discriminative learning loss $L(f)$ for iterative optimization of the target model. The form is as follows:

$$L(f) = \frac{1}{|S_{train}|} \sum_{(e,c)\in S_{train}} \|r(\varphi(e) * f, g_c)\|^2 + \|\lambda f\|^2, \quad (1)$$

where $S_{train} = \{(e_j, c_j)\}_{j=1}^{n}$ is a training set comprising historical frames, including the initial template frame. Each sample frame e_j is paired with the corresponding target center coordinates $c_j \in \mathbb{R}^2$. The feature extraction network φ is used for extracting deep features. The symbol $*$ represents the convolution operation. Through $s = \varphi(e) * f$, foreground and background classification is performed on input sample features, resulting in a target correlation classification response map. The parameters g_c are the desired target scores at each location, set to a Gaussian function centered at c. The function $r(s, g_c)$ computes the residual between s and g_c at each spatial position. By steepest descent with Gauss–Newton iterations, the optimal solution for Equation (1) with respect to f is sought, accomplishing the prediction of the target model.

Leveraging the convolutional layer of the target model, cross-correlation operations are performed on the features of the search region in the test frame, generating corresponding classification response maps for the foreground and background. The target observation center $\hat{P}_t^{(x,y)}$ is determined based on the maximum response position within the response maps.

During the offline training, TCB is optimized by minimizing the classification loss of target models on the test frames. The form is as follows:

$$L_{cls} = \frac{1}{N_{iter}} \sum_{a=0}^{N_{iter}} \sum_{(e,c)\in S_{test}} \left\| l\left(\varphi(e) * f^{(a)}, z_c\right) \right\|^2, \quad (2)$$

where S_{test} is the test set, including the test frame and corresponding target center coordinates. Z_c represents Gaussian pseudo-labels generated based on the target center in the test frame. N_{iter} denotes the number of optimization iterations for the optimal target model, and $f^{(a)}$ is the intermediate model obtained in each iteration. l is a hinge-like residual function.

We reuse the target classification branch of DiMP. We will focus on elaborating the improvement of the regression branch in Section 3.2 and the newly proposed adaptive anti-drift motion constraints in Sections 3.3 and 3.4.

3.2. Attention-Enhanced Regression Branch

For bounding box regression, the baseline tracker DiMP employs a strategy of densely sampling bounding box proposals. Based on the coarse target center localization provided by TCB, a set of bounding box proposals is randomly sampled, and for each proposal, its intersection over union (IoU) with the ground truth is predicted using IoU-Net [48]. During online tracking, the optimization of these proposals is performed by maximizing the predicted IoU. However, in satellite video scenes with substantial interference and targets lacking distinctive features, this dense sampling may introduce numerous hard negative samples, potentially leading to confusion and diminished regression accuracy.

In this section, we improve the bounding box regression branch for satellite video scenes, using an attention-enhanced regression strategy instead of the dense sampling regression strategy. For features extracted from the test frame and template frame, we employ precise ROI pooling (prpool) [48] to obtain the corresponding test patch $X^p \in \mathbb{R}^{C\times H\times W}$ and template patch $Z^p \in \mathbb{R}^{C\times \boldsymbol{h}\times \boldsymbol{w}}$ based on their respective ROI. The patches are then used as inputs for the attention-enhanced regression branch (AERB).

Specifically, the template frame's ROI corresponds to the ground truth region in the template frame. The test frame's ROI is constructed based on the target observation center. Utilizing the height and width of the target from the adjacent frame as size priors,

denoted as w_{prior} and h_{prior}, we create a square ROI in the test frame with a length of $l = \sqrt{(w_{prior} + v) \times (h_{prior} + v)}$, where $v = (w_{prior} + h_{prior})/2$. In contrast to the regression approach involving densely sampling proposals, the ROI-based regression mitigates the introduction of excessive interference in satellite video scenes. Simultaneously, it preserves rich contextual information, which is advantageous for subsequent regression operations.

As illustrated in Figure 2, AERB employs the test patch X^p and template patch Z^p to regress the left top (lt) and right bottom (rb) corners of the target in the test frame. To enhance the regression network's perceptual capability, we introduce a spatial attention mechanism into the regression branch through effective interaction between the template patch and the test patch. We reshape Z^p and X^p into Q, K, and V, where $Q \in \mathbb{R}^{C\times(h\times w)}$ and $K = V \in \mathbb{R}^{C\times(H\times W)}$. By performing matrix multiplication to combine Q and K, we obtain the spatial attention weight matrix for the test patch, denoted as $SM \in \mathbb{R}^{(h\times w)\times(H\times W)}$. We utilize hourglass-like networks HG [49] to process the spatial attention weight matrix and generate spatial attention weight vectors for capturing the local corner features of the target, denoted as $SV_d \in \mathbb{R}^{1\times(H\times W)}$ (visualized in Figure 3), $d \in \{lt, rb\}$. By performing element-wise multiplication, we combine the spatial attention weight vector with V, followed by further reshaping, to obtain enhanced features X_d^p that are sensitive to corner features of the target. The form is as follows:

$$SM = Q^T K, \tag{3}$$

$$X_d^p = \mathrm{Re}[HG_d(SM) \cdot V]. \tag{4}$$

Different channels of X_d^p exhibit varying responses to target corners. To further enhance the precision in localizing target corners, we introduce a channel attention mechanism. This mechanism allows the regression network to adaptively learn which channels are most crucial for corner regression. Following the spatial attention mechanism, we incorporate SENet [50] for channel attention operations.

After enhancing the features using both spatial and channel attention mechanisms, we feed the top-left and bottom-right corner features into basic convolutional networks $Conv_d$. The networks generate heatmaps that represent the probability distribution of corners. The heatmaps are then used to calculate the expected corner coordinates for bounding box regression, enabling the prediction of the target bounding box PB.

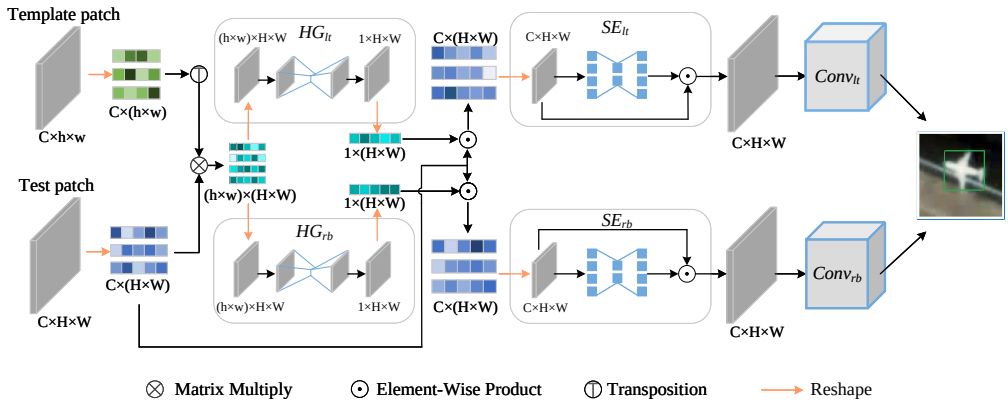

Figure 2. The architecture of AERB. Using the template patch and the test patch as input, the branch introduces spatial attention by considering interactions between patches. HG and SE networks are utilized to enhance corner localization perception in both spatial and channel dimensions.

Figure 3. Visualization of the spatial attention weight vectors for the left top and right bottom corner features of the target.

For AERB, we employ a combination of L_1 and L_2 loss functions between the ground truth bounding box GT and the predicted bounding box PB as the objective function.

$$L_{reg} = \lambda_1 L_1(GT, PB) + \lambda_2 L_2(GT, PB). \tag{5}$$

During the offline training, we utilize the loss function L_{ta} to jointly optimize both TCB and AERB.

$$L_{ta} = \alpha L_{cls} + \beta L_{reg}, \tag{6}$$

where $\alpha = 10^{-2}$ and $\beta = 10^2$ represent the weights for the two types of losses in L_{ta}, while $\lambda_1 = \lambda_2 = 1$ denote the weights for L_1 loss and L_2 loss in L_{reg}.

3.3. Trajectory Distribution Estimation Strategy

In satellite videos, objects often lack distinctive visual features but exhibit rich motion features. In this section, we introduce how the proposed trajectory distribution estimation module (TDEM) leverages the motion features of targets for trajectory distribution estimation, compensating for the deficiency of visual features.

TDEM takes historical trajectory segments of the tracked target as input rather than the entire trajectory. During satellite observation, the movement mode of the target may change, such as turning, accelerating, slowing down, etc. Modeling target motion using the entire trajectory may not promptly reflect these changes in motion patterns, leading to unreliable estimates. Estimating using trajectory segments allows for a more responsive reflection of the object's evolving motion patterns, thereby enhancing short-term prediction accuracy. We use $T(t{-}n{:}t{-}1)$ to denote the historical trajectory segment of the target in Frame t.

$$T(t-n:t-1) = \left\{ BB_{t-n}^{(x,y)}, BB_{t-n+1}^{(x,y)}, \cdots, BB_{t-1}^{(x,y)} \right\}, \tag{7}$$

where $BB_{t-1}^{(x,y)}$ represents the center position (x, y) of the target in Frame $t-1$.

As shown in Figure 4, we input $T(t{-}n+1{:}t{-}1)$ in sequential form into the LSTM network. By extracting interdependencies between different time steps within the trajectory segment, LSTM generates the hidden state $H_t \in \mathbb{R}^{1\times 128}$ at time step t, which serves as the local feature for estimating the trajectory distribution of the target in Frame t. We further utilize a fully connected layer to map this local feature to the space of target trajectory regression, resulting in an estimate of the target trajectory, denoted as $P_t^{(x,y)}$. We combine the prior information from the bounding box $BB_{t-1}^{(w,h)}$ with $P_t^{(x,y)}$ and use them as compensation to refine the predictions made by AERB for the target bounding box.

$$P_t^{(x,y)} = w_1(\mathrm{LSTM}(T(t-n:t-1))) + b_1, \tag{8}$$

where w_1 and b_1 represent the parameters of the fully connected layer (FC).

In our approach, we extensively utilize the dynamic dependencies present in historical trajectories. Instead of merely predicting the target's trajectory distribution at the current time step for compensatory corrections during tracking, our method also incorporates trajectory predictions from multiple frames. This approach establishes a comprehensive distribution of the target's motion trends, aiding in the evaluation of the tracking status.

After obtaining the estimated target location at time step t, it is concatenated sequentially with $T(t-n+1{:}t-1)$ and then inputted into TDEM. The process is repeated to estimate the target trajectory at time steps $t+1$, $t+2$, ..., $t+m-1$, thereby achieving multi-frame trajectory predictions. We model the trajectory distribution at a single time step as a two-dimensional Gaussian distribution $N_i(\mu_i, \Sigma)$, $i \in [0, m)$:

$$\mu_i = \begin{bmatrix} x_{t+i} \\ y_{t+i} \end{bmatrix}, \Sigma = \begin{bmatrix} \varepsilon r & \mathbf{0} \\ \mathbf{0} & \varepsilon r \end{bmatrix}, \tag{9}$$

where (x_{t+i}, y_{t+i}) represents the estimated target trajectory at time step $t+i$, while $r = \sqrt{w_{res} \times h_{res}}$, w_{res} and h_{res} represent the size of the classification response map output by TCB. ε is a distribution hyperparameter in this context.

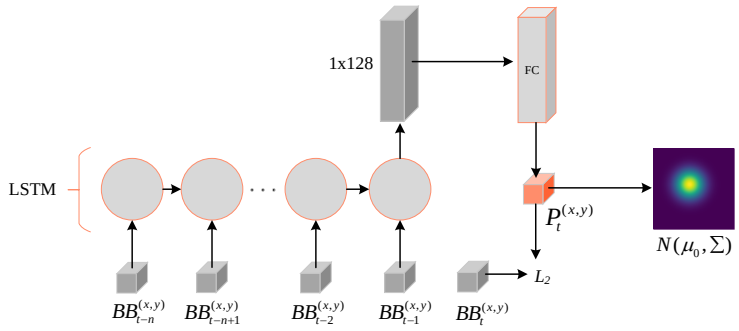

Figure 4. The architecture of TDEM. Motion features are extracted from the historical trajectory of the target using LSTM to estimate the target's trajectory distribution at step t.

To capture the motion trends of the target, we combine multiple frame trajectory predictions and employ a mixture Gaussian distribution model to describe the distribution of motion trends.

$$P(x,y) = \sum_{i=0}^{m-1} \frac{N_i(x,y|\mu_i,\Sigma)}{m}. \tag{10}$$

During the training phase, we supervise the training of TDEM by calculating L_2 loss between the estimated target trajectory $P_t^{(x,y)}$ and the ground truth in Frame t. In Section 3.4, we will describe how to utilize the target motion trends to implement our proposed anti-drift strategy, thereby enhancing the robustness of the tracker.

3.4. Anti-Drift Strategy

In satellite video tracking, tracking drift greatly affects accuracy, making it a critical issue. To address this, we have developed an anti-drift module (ADM), which leverages the target motion trend distribution produced by TDEM. ADM integrates motion constraints into the tracking algorithm via a learnable drift detection strategy, thus improving the tracker's robustness and generalization ability. The implementation process is as follows:

We model the tracker observation distribution by using the target observation center $\hat{P}_t^{(x,y)}$ on the classification response map of the test frame as the mean vector $\hat{\mu}$ of a two-dimensional Gaussian distribution $\hat{P}(x,y)$, with Σ as the covariance matrix.

$$\hat{P}(x,y) = N(x,y \mid \hat{\mu}, \Sigma). \tag{11}$$

The motion trend distribution of the target over m consecutive frames is fused with the tracker observation distribution to obtain the discrepancy distribution $\bar{P}(x, y)$.

$$\bar{P}(x, y) = P(x, y) - \hat{P}(x, y). \tag{12}$$

We utilize a grid-based uniform sampling strategy to discretize and reshape the discrepancy distribution according to the response map size, resulting in a discrete differential distribution feature description $D \in \mathbb{R}^{1\times(h_{res}\times w_{res})}$. The description is then inputted into ADM composed of the multi-layer perceptron. Through two linear layers, the module predicts the current tracking state probability distribution SP. This process can be described as follows:

$$SP = \text{Softmax}(w_3[\text{RELU}(w_2 D + b_2) + b_3]), \tag{13}$$

where w_2 and b_2, w_3 and b_3 represent the parameters of the fully connected layers in the multi-layer perceptron.

During the training of ADM, we label the tracking state based on whether the target observation center $\hat{P}_t^{(x,y)}$ is inside the ground truth bounding box. If it is inside the ground truth bounding box, we consider the tracking state as normal; otherwise, we consider it as abnormal. Using this labeling scheme, we calculate the loss for ADM, which is defined as the cross-entropy loss between the label distribution and the predicted probability distribution. During the online tracking, TDEM requires n frames as input for motion modeling. After the initial n frames, we use ψ as an indicator to determine whether to employ motion features for compensatory correction in the tracking process.

$$\psi = \text{Argmax}(SP). \tag{14}$$

Expanding upon trajectory distribution estimation, our proposed ADM incorporates historical trajectory into the target tracking process through drift detection. The approach enables the tracker to accurately identify and rectify drift-related issues, regardless of whether they are caused by interference or occlusion. Algorithm 1 summarizes the proposed motion constrained tracking.

Algorithm 1 The procedure of motion constraint for tracking.

Input:

$T(t-n:t-1)$: The historical trajectory segment of the target;
$\hat{P}_t^{(x,y)}$: The target observation center in Frame t;
Σ: The covariance matrix of the trajectory distribution;
PB: The prediction of the target bounding box by the AERB;
$BB_{t-1}^{(w,h)}$: The prior information about the bounding box;

Output:

TR_t: The tracking result of Frame t;

$\hat{P}(x,y) \xleftarrow{\text{Equation (11)}} \left\{\hat{P}_t^{(x,y)}, \Sigma\right\}$
$T = T(t-n, t-1)$
$TP = \{\}$
for $i < m$ **do**
  $P_{t+i}^{(x,y)} \xleftarrow{\text{Equation (8)}} T$
  $TP = TP \cup \left\{P_{t+i}^{(x,y)}\right\}$
  $T \xleftarrow{\text{Concatenate}} \left\{T(t-n+1+i:t-1+i), P_{t+i}^{(x,y)}\right\}$
end for
$CB = \left[P_t^{(x,y)}, BB_{t-1}^{(w,h)}\right]$
$P(x,y) \xleftarrow{\text{Equation (10)}} \left\{TP, \Sigma\right\}$
$\psi \xleftarrow{\text{Equations (12)–(14)}} \left\{P(x,y), \hat{P}(x,y)\right\}$
if ψ **then**
  $TR_t = CB$
else
  $TR_t = PB$
end if
return Output

4. Experiments and Results Analysis

4.1. Experimental Settings

4.1.1. Datasets

We employ the SatVideoDT challenge dataset [51], which was collected from the Jilin-1 video satellite, for both training and testing. We also report results on the SaSOT benchmark dataset [52]. The SaSOT dataset consists of data from three video satellites: Jilin-1, Skybox, and Carbonite-2 [6]. Due to the partial overlap between SaSOT and SatVideoDT, we remove the repeated video sequences from the SatVideoDT dataset, resulting in 8301 sequences for training and 1126 sequences for testing. The SaSOT dataset remains unchanged.

The SaSOT dataset includes four classes of tracking objects: cars, trains, airplanes, and ships, with an average of 263 frames per video sequence. The dataset includes 11 challenging attributes, as listed in Table 1. The distribution of object sizes in the dataset spans a wide range, from 21 to 780,605 pixels. The SatVideoDT dataset primarily comprises small-sized objects, such as vehicles, with over 98% of the bounding box sizes being less than 100 pixels, aiming at evaluating the performance of tracking tiny objects in satellite videos. The average video sequence length is 217 frames.

Table 1. Definitions of the 11 challenging attributes in the SatSOT dataset.

Attribute	Definition
BC	background clutter: the background has similar appearance to the target
IV	illumination variation: the illumination of the target region changes significantly
LQ	low quality: the image is in 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 temporally 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 occurs by the shaking of satellite camera
DEF	deformation: non-rigid object deformation
ARC	aspect ratio change: the ratio of the box aspect ratio of the first and the current frame is outside the range [0.5, 2]

4.1.2. Evaluation Metrics

We follow the OTB paradigm [53], utilizing a precision plot and a success plot to demonstrate the performance of trackers.

The precision plot evaluates the performance of a tracker using the center location error (CLE) between the predicted target center (x_p, y_p) and the ground truth center (x_g, y_g).

$$\mathrm{CLE} = \sqrt{(x_p - x_g)^2 + (y_p - y_g)^2}. \tag{15}$$

In the precision plot, the horizontal axis denotes the CLE threshold, and the vertical axis illustrates the percentage of frames in the video sequences where the CLE of the predicted target center is below the specified threshold. As objects in satellite videos are typically small, we establish the rankings of the precision (Prec.) for trackers based on their performance at the CLE threshold of five pixels.

The success plot evaluates the performance of a tracker using the IoU between the predicted bounding box (A_p) and the ground truth bounding box (A_g).

$$\mathrm{IoU} = \frac{A_p \bigcap A_g}{A_p \bigcup A_g}. \tag{16}$$

In the success plot, the horizontal axis represents the IoU threshold, and the vertical axis represents the proportion of frames in the video sequences where the IoU of the predicted bounding box is greater than the specified threshold. The success rate (Succ.) of trackers is ranked based on the area under the curve of the success plot.

Additionally, we utilize the frames per second (FPS) metric to illustrate the speed of trackers. A higher FPS value indicates faster tracking speed.

4.1.3. Implementation Details

We implement our tracker using the PyTorch deep learning framework and train it on an Ubuntu 20.04 platform equipped with an NVIDIA RTX A6000 GPU. For feature extraction, we utilize the ResNet50 [54] pre-trained on ImageNet [55]. The training process of our tracker comprised three stages.

In the first stage, we jointly train TCB and AERB for 50 epochs. The TCB learning rates are the same as in the baseline tracker, and the AERB learning rate is set to 1×10^{-3}. In the second stage, we sample continuous sequences of n + 1 frames from video sequences as training samples for training TDEM. The learning rate is set to 1×10^{-3}, and this stage involves 50 epochs of training. In the third stage, we freeze the weights of the models trained in the preceding two stages. Similar to the second stage, we sample continuous sequences of n + 1 frames as training samples to train ADM. The learning rate is set to 1×10^{-3}, and we conduct training for 40 epochs. We use ADAM [56] with a learning rate decay of 0.2 every 15th epoch for every training stage.

The online tracking experiments are conducted on an Ubuntu 18.04 platform with an NVIDIA GeForce GTX 1060 GPU. In our tracking protocol, if the trajectory estimation results serve as the final tracking outcome for m consecutive frames, it is inferred that the target has disappeared from the search area. In such cases, we expand the search area to three times its original size to search for the target.

4.2. Ablation Study

4.2.1. Study on Key Components

The proposed tracker consists of three components: TCB, AERB, and MCB. To evaluate the impact of each component on performance, we conducted ablation experiments on the SatSOT dataset, and the baseline method is DiMP. The results are presented in Table 2.

Table 2. Ablation study of TCB, AERB, and MCB on SatSOT, with the best results highlighted in red.

Tracker	TCB	AERB	MCB	Prec. (%)	Succ. (%)
Baseline	✓	-	-	56.7	41.9
TCB + AERB	✓	✓	-	61.6	46.2
TCB + MCB	✓	-	✓	60.5	42.9
TCB + AERB + MCB	✓	✓	✓	66.3	49.0

The ablation experiments demonstrate that both AERB and MCB independently contribute to improving the performance of the baseline tracker. Specifically, AERB leads to an improvement of 4.9% in precision and 4.3% in success rate. The results indicate that in satellite video scenes, the attention-enhanced regression strategy in the region of interest employed by AERB is superior to the dense sampling regression strategy in the baseline tracker. Furthermore, MCB, which constrains the tracking process using the historical trajectory information, results in a 3.8% improvement in precision and a 1% improvement in success rate. The results suggest that introducing historical trajectory information into the tracking process is effective in enhancing tracking performance.

By incorporating both AERB and MCB into the tracker, we achieve a 9.6% improvement in precision and a 7.1% improvement in success rate. The experiments highlight the crucial roles played by AERB and MCB in our tracker.

4.2.2. Study on Attention-Enhanced Regression Branch

In the AERB, two types of attention mechanisms are incorporated: spatial attention and channel attention. We facilitate interaction between the test patch and the template patch by introducing spatial attention during the bounding box regression. This enhances the regression network's ability to perceive the corners of targets, and the results in Table 3 demonstrate that this mechanism leads to effective performance gains. Furthermore, we employ channel attention operations to capitalize on the significant variations in corner response across different feature channels. This further augments the network's capability to precisely locate the target corners. As indicated by the results in Table 3, the combination strategy significantly improves both the precision and success rate of the tracker.

Table 3. Ablation study of spatial (S) and channel (C) attention in the AERB on SatSOT, with the best results highlighted in red.

Tracker	S	C	Prec. (%)	Succ. (%)
TCB + MCB	-	-	60.5	42.9
TCB + MCB + S	✓	-	61.7	45.2
TCB + MCB + S + C	✓	✓	66.3	49.0

4.2.3. Study on Motion Constraint Branch

The proposed MCB consists of two main components: TDEM and ADM. TDEM utilizes historical trajectory information to estimate trajectory distribution, while ADM

utilizes TDEM to detect tracking drift and implement motion constraints and compensation during the tracking process. The effectiveness of TDEM and ADM is demonstrated through ablation experiments conducted on the SatSOT dataset. Without ADM, we utilize peak responses in the response map outputted by TCB to evaluate the current tracking state. When the value of the peak response falls below a given threshold, indicating tracking uncertainty, TDEM is employed for motion compensation. The results in Table 4 indicate that the improvement in tracking performance is influenced by the specified threshold. The best performance is achieved when the threshold is set to 0.3. Furthermore, the combination of ADM and TDEM brings a more significant improvement in tracker performance compared to introducing TDEM alone, further confirming the effectiveness of MCB.

Table 4. Ablation study of TDEM and ADM in the MCB on SatSOT, where "/0.1", "/0.2", "/0.3", and "/0.4" denote setting the response threshold to 0.1, 0.2, 0.3, and 0.4, respectively, with the best results highlighted in red.

Tracker	TDEM	ADM	Prec. (%)	Succ. (%)
TCB + AERB	-	-	61.6	46.2
TCB + AERB + TDEM/0.1	✓	-	60.9	45.6
TCB + AERB + TDEM/0.2	✓	-	61.3	45.9
TCB + AERB + TDEM/0.3	✓	-	63.5	47.2
TCB + AERB + TDEM/0.4	✓	-	62.2	46.3
TCB + AERB + MCB	✓	✓	66.3	49.0

4.2.4. Study on Trajectory Distribution Estimation Strategy

In the process of constructing the motion model, we utilize trajectory segments to extract historic target motion features and predict the distribution of the current frame's target trajectory. Simultaneously, we employ multi-frame trajectory predictions to model the trend distribution of the target motion. We conduct ablation experiments on the length of trajectory segments (denoted as n) and the number of frames (denoted as m) for multi-frame predictions, with the results presented in Figure 5.

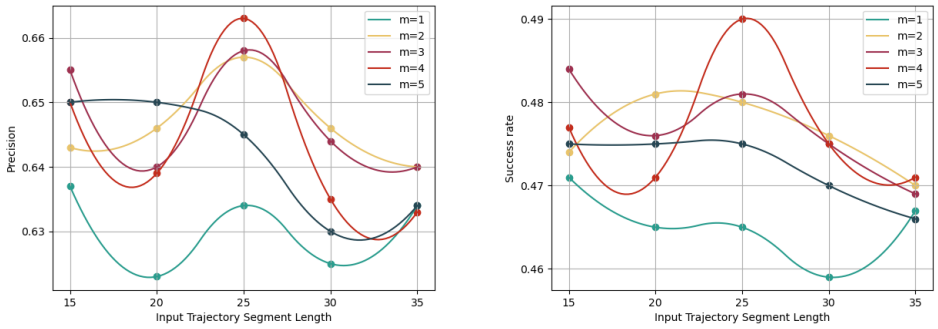

Figure 5. Study on the length of input trajectory segment and the number of frames for multi-frame predictions on SatSOT.

The proposed tracker's performance initially improves and then diminishes with increasing trajectory segment length when the number of prediction frames is fixed. This suggests that in satellite video scenes, there is limited reliance on long-term historical trajectories for target motion. Utilizing long-term trajectories to constrain target tracking may lead to performance degradation due to error accumulation and possible motion mode change. We also conduct ablation experiments on the number of prediction frames with a fixed trajectory segment length. Specifically, when the number of prediction frames is set to 1, indicating the absence of a multi-frame trajectory prediction strategy, the results from the ablation experiments show that using the multi-frame trajectory prediction strategy generally outperforms not using it. When $n = 25$ and $m = 4$, the network achieves the best results.

4.2.5. Study on Distribution Hyperparameter

In the designed motion constraint branch, the distribution parameter ε needs to be set for both the motion trend distribution and the observation distribution of the target. According to the experimental results in Table 5, setting this parameter to 0.05 yields the best tracking performance. With the optimal parameter setting, we also evaluated the tracking efficiency. The tracker achieves 25.05 FPS, thereby meeting the real-time requirements of satellite video processing.

Table 5. Study of the distribution parameter in MCB on SatSOT, with the best results highlighted in red.

ε	Prec. (%)	Succ. (%)	FPS
0.07	65.8	48.3	25.29
0.06	64.6	47.6	25.49
0.05	66.3	49.0	25.05
0.04	63.9	47.2	24.85
0.03	64.1	47.4	24.84

4.3. Results and Analysis

4.3.1. Overall Results

We conducted comparative experiments with representative tracking algorithms in satellite video scenes, including conventional correlation filter trackers (KCF [12], STRCF [57], and ECO [26]), deep learning-based correlation filter trackers (ATOM [58], PrDiMP [28], and SuperDiMP [29]), Siamese network trackers (SiamBAN [34], SiamCAR [33], and CGACD [35]), Transformer trackers (STARK [15], and Mixformer [16]), a specialized tracker for satellite video (CFME [42]), and the baseline tracker (DiMP). We fine-tuned deep learning-based tracking methods on the satellite video dataset using their respective open-sourced models.

Table 6 presents the precision and success rates of various trackers on the SatSOT and SatVideoDT datasets. Our proposed tracker, benefiting from improvements in the bounding box regression stage and the introduction of motion information, significantly outperforms various trackers on both datasets. Among the deep learning trackers that use Resnet50 as the feature extractor, our tracker achieves a significant gain of 7.7% in precision and 5.8% in success rate compared to the second-based SiamCAR on the SatSOT dataset. On the SatVideoDT dataset, our tracker outperforms the second-best SuperDiMP with a 6.7% improvement in precision and a 5.0% improvement in success rate. Additionally, our proposed tracker exhibits a leading performance when compared to the Transformer-based trackers (STARK and Mixformer).

Compared to CFME on the SatSOT dataset, our tracker shows an improvement of 11.7% in precision and 7.2% in success rate. Nevertheless, conventional correlation-based trackers still demonstrate competitive precision in target localization compared to deep learning-based trackers in satellite video scenes. Specifically, the STRCF tracker achieves the second-highest tracking precision on the SatVideoDT dataset, but its performance in estimating the size of the bounding box is poor at a low success rate due to the lack of effective regression strategies.

Table 6. Comparison of SatSOT and SatVideoDT; the top three results are highlighted in red, green, and blue. For methods, CF: traditional correlation filter methods, DCF: Deep learning-based correlation filter methods, SN: Siamese network methods, and TF: Transformer methods. For features, HoG: HoG features, CNN: Convolutional networks, TF: Transformer networks.

Tracker	Method	Feature	SatSOT		SatVideoDT	
			Prec. (%)	Succ.(%)	Prec. (%)	Succ. (%)
KCF	CF	HOG	21.5	22.2	8.7	3.2
STRCF	CF	HOG	52.3	36.8	56.1	20.6
ECO	CF	CNN	55.2	36.8	44.8	15.0
CFME	CF	HOG	54.6	41.8	35.6	19.0
ATOM	DCF	CNN	55.6	39.4	40.9	19.0
DiMP	DCF	CNN	56.7	41.9	52.9	26.2
PrDiMP	DCF	CNN	49.8	35.8	44.0	20.6
SuperDiMP	DCF	CNN	57.3	42.6	54.2	27.6
SiamBAN	SN	CNN	57.2	41.6	50.5	23.8
SiamCAR	SN	CNN	58.6	43.2	52.8	26.4
CGACD	SN	CNN	56.5	42.2	38.1	16.4
STARK	TF	CNN	51.1	36.6	47.4	25.0
Mixformer	TF	TF	54.0	43.6	51.9	27.7
Ours	DCF	CNN	66.3	49.0	60.9	32.6

4.3.2. Against Different Challenges

In satellite video scenes, trackers face challenges, such as similar object interference, background clutter, low quality imaging, partial occlusion, etc. We evaluate different trackers on various challenge attributes described by the SatSOT dataset. We generate success plots and precision plots for the trackers on each challenge.

As shown in Figure 6, the proposed tracker ranks first in precision under six challenge attributes: background clutter, partial occlusion, tiny object, similar object, rotation, and low quality. As illustrated in Figure 7, the proposed tracker ranks first in success rate under seven challenge attributes: background clutter, partial occlusion, tiny object, similar object, rotation, low quality, and illumination variation. Compared to other trackers, the proposed tracker demonstrates strong performance in both precision and success rate when facing complex challenges.

However, the proposed tracker exhibits limitations in handling background jitter, deformation, aspect ratio changes, and full occlusion challenges. For deformation and aspect ratio change, we find that these challenges mainly occurred with the category of trains in satellite video scenes. The long shape of a train leads to extreme aspect ratios in its bounding box. When a train makes turns and changes shape, the size of the bounding box also changes significantly. As the proposed tracker does not incorporate specific adaptation strategies for these challenges, it may struggle to accurately represent train features, leading to tracking failures. Among the trackers that use Resnet50 as the feature extractor, the proposed tracker demonstrates relatively good performance in handling deformation and aspect ratio change.

In terms of handling background jitter and full occlusion challenges in satellite video scenes, our proposed tracker ranks second only behind the top-performing CFME, which also utilizes a motion model. The reason could be that introducing motion constraints into the tracking process through a learnable approach often requires a substantial amount of training data to learn how to model target motion patterns when background jitter and full occlusion challenges occur. If these factors are not captured adequately during training, it can limit the performance of the tracker. Nevertheless, our tracker is not constrained by specific motion assumptions, theoretically offering better adaptability to complex scenes. In future works, we plan to simulate such challenging scenes for training, which will address the lack of enough case samples and benefit the learnable approach.

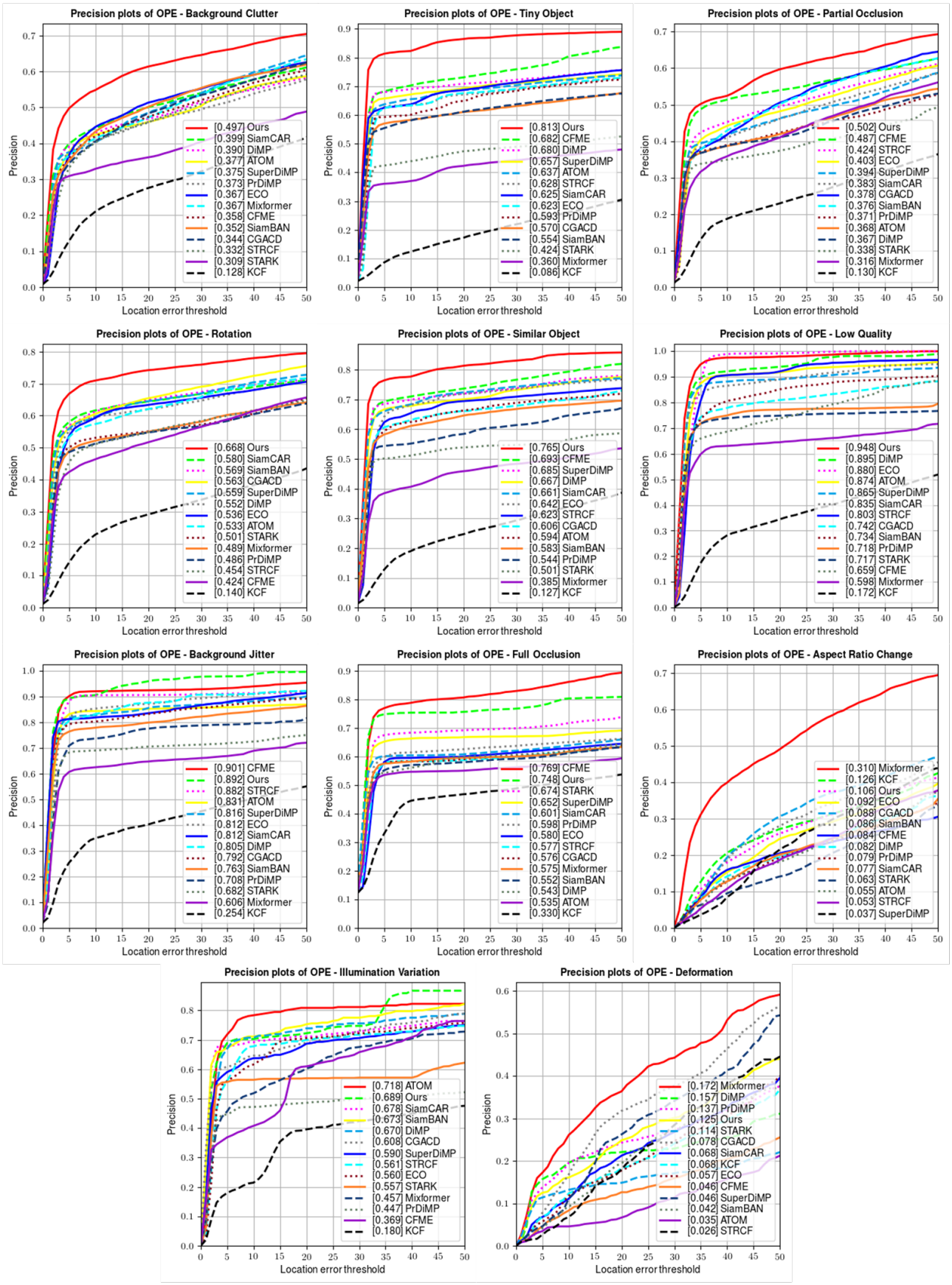

Figure 6. Precision plots of trackers across 11 challenge attributes.

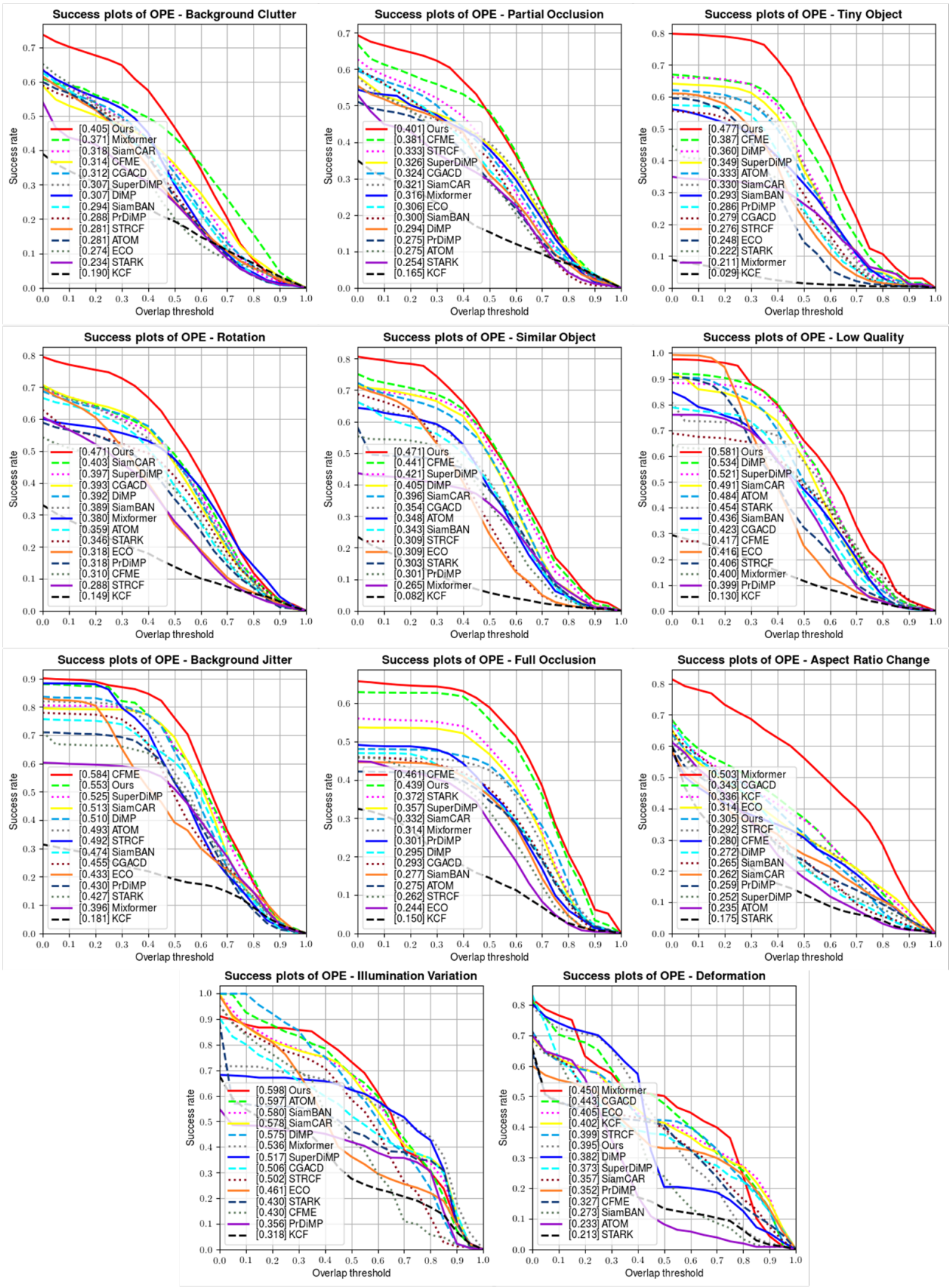

Figure 7. Success plots of trackers across 11 challenge attributes.

4.3.3. Qualitative Results

We present the tracking results for eight satellite videos from the SatSOT dataset, with their challenge attributes detailed in Table 7. For visual comparison, we selected five representative trackers: SiamCAR (a Siamese network tracker), ECO (a conventional correlation filter tracker), CFME (a tracker specifically designed for satellite videos), Mixformer (a Transformer tracker), and SuperDiMP (a deep learning-based correlation filter tracker).

Table 7. Selected video sequences with challenge attributes. (FOC: full occlusion, POC: partial occlusion, ROT: rotation, TO: tiny object, SOB: similar object, BC: background clutter, ARC: aspect ratio change, DEF: deformation, LQ: low quality.)

Video Name	Challenge Attributes
Car_24	FOC, POC, ROT
Car_34	FOC, TO, SOB, ROT, BC
Car_40	TO, POC, ROT, BC
Car_53	SOB, TO
Plane_09	ROT
Train_08	BC, ARC
Train_01	BC, DEF, ARC, ROT
Car_50	BC, ROT, LQ, POC, SOB

The video sequences Car_53, Car_34, and Car_40 shown in Figure 8 reflect the representation of small targets in satellite videos. Most trackers fail to effectively extract distinctive appearance features for such small targets, especially in background clutter. Taking Car_53, for example, it can be observed that trackers like SiamCAR and CFME are affected by the small target size and drift to the surrounding road or nearby similar objects, leading to tracking failures. In contrast, our tracker benefits from improvements in the bounding box regression strategy and can more stably identify the target from various types of background interference compared to other trackers. In the video sequences Car_24, Car_34, and Car_40, there are brief disappearances of the targets due to being obscured by overpasses or building shadows, especially in Car_24, where the target is occluded twice around the 70th and 210th frames. We observe that our tracker effectively identifies the occlusion and disappearance of the target in Car_24, thanks to the assistance of the motion model. Unlike other trackers, our tracker can resume tracking when the target reappears. In the Plane_09 video sequence, which represents the rotation challenge in satellite videos, several trackers fail. Due to the timely updates of the discriminative target model, our tracker can capture changes in target features during the rotation process, achieving relatively accurate tracking.

Although the proposed tracker has shown excellent performance facing various scenes and challenges, it occasionally fails in some challenging scenarios. In Figure 9, we present visualized results of our tracker when it faces the ARC and DEF challenges in the Train_01 sequence. It can be seen that although we have not set specific strategies for these two attributes, our tracker demonstrates relatively stable performance compared to other trackers. However, when facing extreme ARC and DEF challenges after the 90th frame in the Train_01 sequence, all trackers including ours fail to track. This indicates that ARC and DEF attributes in the satellite video scene remain a challenging task.

As shown in video sequence Car_50 in Figure 9, the low-quality attribute makes it difficult for trackers to extract distinguishable target features, leading to an unstable tracking process where almost all trackers lose track of the target after the 200th frame. To mitigate this issue, better feature extraction or image enhancement strategies are needed to assist trackers in handling low-quality scenes.

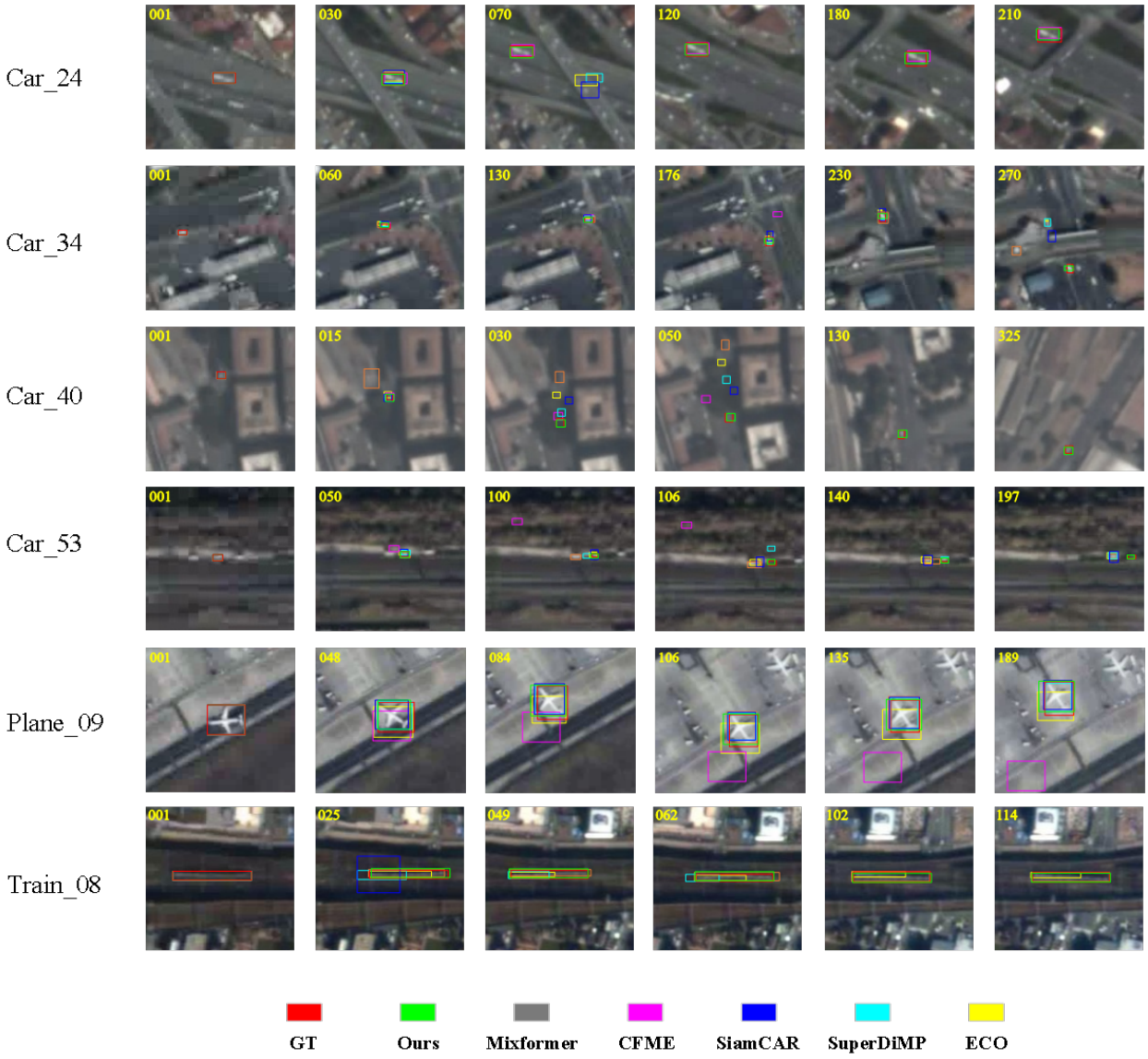

Figure 8. Tracking visualization on the Car_24, Car_34, Car_40, Car_53, Plane_09, and Train_08 video sequences of the SatSOT dataset. The yellow number at the top left of the image represents the video frame number.

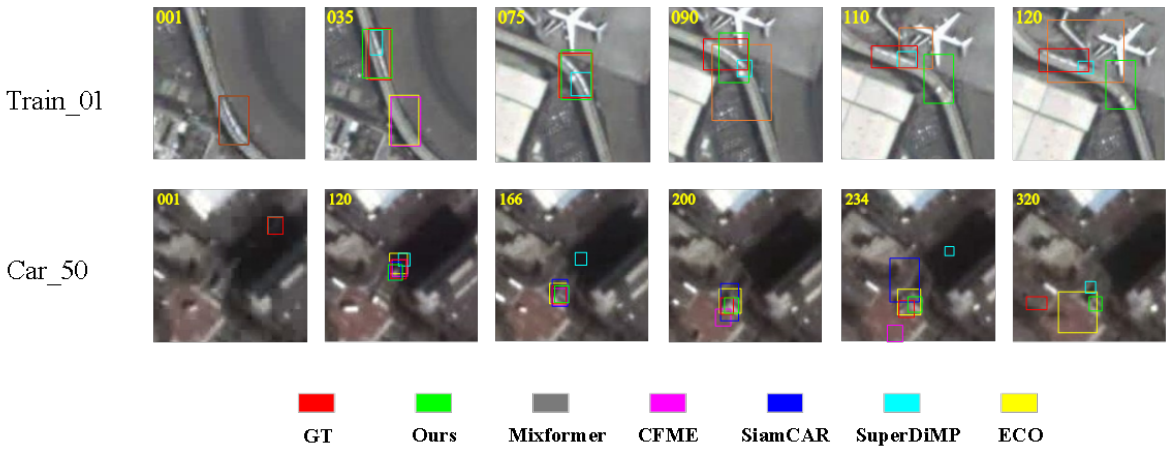

Figure 9. Failure cases of the proposed tracker on the Train_01, and Car_50 of the SatSOT dataset. The yellow number at the top left of the image represents the video frame number.

5. Conclusions

In this article, we propose a correlation filter-based learnable tracker for satellite videos. The tracker utilizes an attention-enhanced bounding box regression branch to improve the network's ability to distinguish between the target and complex backgrounds during the regression stage, and accurately locate and regress target corners. Additionally, the motion constraints branch that includes the trajectory distribution estimation module (TDEM) and the anti-drift module (ADM) is proposed to assist the tracking process, to effectively constrain and compensate tracking drift. We conducted extensive experiments on two satellite video datasets, SatSOT and SatVideoDT, and compared our method to state-of-the-art trackers including common ones and those specially designed for satellite videos. Our algorithm demonstrates superior performance, as confirmed by the experimental results on the SatSOT and SatVideoDT datasets. In future work, we plan to explore more effective strategies for tracking trains that undergo extreme aspect ratio changes and non-rigid deformations, as well as for addressing low-quality issues in satellite videos.

Author Contributions: Methodology, J.F.; resources, S.J.; writing—original draft preparation, J.F.; writing—review and editing, S.J.; supervision, S.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (grant No. 42171430) and the State Key Program of the National Natural Science Foundation of China (grant No. 42030102).

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: https://doi.org/10.1109/TGRS.2022.3140809, https://doi.org/10.1109/ICPR56361.2022.9956153.

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

References

1. Tsakanikas, V.; Dagiuklas, T. Video surveillance systems-current status and future trends. *Comput. Electr. Eng.* **2018**, *70*, 736–753. [CrossRef]
2. Singha, J.; Roy, A.; Laskar, R.H. Dynamic hand gesture recognition using vision-based approach for human–computer interaction. *Neural Comput. Appl.* **2018**, *29*, 1129–1141. [CrossRef]
3. Grigorescu, S.; Trasnea, B.; Cocias, T.; Macesanu, G. A survey of deep learning techniques for autonomous driving. *J. Field Robot.* **2020**, *37*, 362–386. [CrossRef]
4. Wilson, D.; Alshaabi, T.; Van Oort, C.; Zhang, X.; Nelson, J.; Wshah, S. Object Tracking and Geo-Localization from Street Images. *Remote Sens.* **2022**, *14*, 2575. [CrossRef]
5. d'Angelo, P.; Kuschk, G.; Reinartz, P. Evaluation of Skybox Video and Still Image products. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* **2014**, *XL-1*, 95–99. [CrossRef]
6. Cheng, G.; Han, J.; Lu, X. Remote Sensing Image Scene Classification: Benchmark and State of the Art. *Proc. IEEE* **2017**, *105*, 1865–1883. [CrossRef]
7. Cui, K.; Xiang, J.; Zhang, Y. Mission planning optimization of video satellite for ground multi-object staring imaging. *Adv. Space Res.* **2018**, *61*, 1476–1489. [CrossRef]
8. Xian, Y.; Petrou, Z.I.; Tian, Y.; Meier, W.N. Super-Resolved Fine-Scale Sea Ice Motion Tracking. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 5427–5439. [CrossRef]
9. Melillos, G.; Themistocleous, K.; Papadavid, G.; Agapiou, A.; Prodromou, M.; Michaelides, S.; Hadjimitsis, D.G. Integrated use of field spectroscopy and satellite remote sensing for defence and security applications in Cyprus. In Proceedings of the Conference on Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XXI, Baltimore, MD, USA, 18–21 April 2016. [CrossRef]
10. Alvarado, S.T.; Fornazari, T.; Cóstola, A.; Morellato, L.P.C.; Silva, T.S.F. Drivers of fire occurrence in a mountainous Brazilian cerrado savanna: Tracking long-term fire regimes using remote sensing. *Ecol. Indic.* **2017**, *78*, 270–281. [CrossRef]
11. Bolme, D.S.; Beveridge, J.R.; Draper, B.A.; Lui, Y.M. Visual object tracking using adaptive correlation filters. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 2544–2550. [CrossRef]
12. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. High-Speed Tracking with Kernelized Correlation Filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 583–596. [CrossRef]

13. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H.S. Fully-Convolutional Siamese Networks for Object Tracking. In Proceedings of the Computer Vision—ECCV 2016 Workshops, Amsterdam, The Netherlands, 8–16 October 2016; pp. 850–865. [CrossRef]
14. Li, B.; Wu, W.; Wang, Q.; Zhang, F.; Xing, J.; Yan, J. SiamRPN++: Evolution of Siamese Visual Tracking With Very Deep Networks. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 4277–4286. [CrossRef]
15. Yan, B.; Peng, H.; Fu, J.; Wang, D.; Lu, H. Learning Spatio-Temporal Transformer for Visual Tracking. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 10428–10437. [CrossRef]
16. Cui, Y.; Jiang, C.; Wang, L.; Wu, G. MixFormer: End-to-End Tracking with Iterative Mixed Attention. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 13598–13608. [CrossRef]
17. Bhat, G.; Danelljan, M.; Van Gool, L.; Timofte, R. Learning Discriminative Model Prediction for Tracking. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6181–6190. [CrossRef]
18. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
19. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. Exploiting the Circulant Structure of Tracking-by-Detection with Kernels. In Proceedings of the European Conference on Computer Vision (ECCV), Florence, Italy, 7–13 October 2012; pp. 702–715. [CrossRef]
20. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; pp. 886–893. [CrossRef]
21. Li, Y.; Zhu, J. A Scale Adaptive Kernel Correlation Filter Tracker with Feature Integration. In Proceedings of the Computer Vision—ECCV 2014 Workshops, Zurich, Switzerland, 6–12 September 2015; pp. 254–265. [CrossRef]
22. van de Weijer, J.; Schmid, C.; Verbeek, J.; Larlus, D. Learning Color Names for Real-World Applications. *IEEE Trans. Image Process.* **2009**, *18*, 1512–1523. [CrossRef]
23. Galoogahi, H.K.; Fagg, A.; Lucey, S. Learning Background-Aware Correlation Filters for Visual Tracking. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 1144–1152. [CrossRef]
24. Danelljan, M.; Robinson, A.; Khan, F.S.; Felsberg, M. Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 472–488. [CrossRef]
25. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the 3rd International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015. [CrossRef]
26. Danelljan, M.; Bhat, G.; Khan, F.S.; Felsberg, M. ECO: Efficient Convolution Operators for Tracking. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6931–6939. [CrossRef]
27. Valmadre, J.; Bertinetto, L.; Henriques, J.; Vedaldi, A.; Torr, P.H.S. End-to-End Representation Learning for Correlation Filter Based Tracking. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5000–5008. [CrossRef]
28. Danelljan, M.; Van Gool, L.; Timofte, R. Probabilistic Regression for Visual Tracking. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 7181–7190. [CrossRef]
29. Danelljan, M.; Bhat, G.; Mayer, C.; Paul, M. pytracking. Available online: https://github.com/visionml/pytracking (accessed on 21 January 2024).
30. Li, B.; Yan, J.; Wu, W.; Zhu, Z.; Hu, X. High Performance Visual Tracking with Siamese Region Proposal Network. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8971–8980. [CrossRef]
31. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
32. Zhu, Z.; Wang, Q.; Li, B.; Wu, W.; Yan, J.; Hu, W. Distractor-Aware Siamese Networks for Visual Object Tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 103–119. [CrossRef]
33. Guo, D.; Wang, J.; Cui, Y.; Wang, Z.; Chen, S. SiamCAR: Siamese Fully Convolutional Classification and Regression for Visual Tracking. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 6268–6276. [CrossRef]
34. Chen, Z.; Zhong, B.; Li, G.; Zhang, S.; Ji, R.; Tang, Z.; Li, X. SiamBAN: Target-Aware Tracking With Siamese Box Adaptive Network. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 5158–5173. [CrossRef] [PubMed]
35. Du, F.; Liu, P.; Zhao, W.; Tang, X. Correlation-Guided Attention for Corner Detection Based Visual Tracking. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 6835–6844. [CrossRef]

36. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. In Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
37. Chen, X.; Yan, B.; Zhu, J.; Wang, D.; Yang, X.; Lu, H. Transformer Tracking. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 8122–8131. [CrossRef]
38. Cao, Z.; Huang, Z.; Pan, L.; Zhang, S.; Liu, Z.; Fu, C. TCTrack: Temporal Contexts for Aerial Tracking. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 14778–14788. [CrossRef]
39. Du, B.; Sun, Y.; Cai, S.; Wu, C.; Du, Q. Object Tracking in Satellite Videos by Fusing the Kernel Correlation Filter and the Three-Frame-Difference Algorithm. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 168–172. [CrossRef]
40. Liu, Y.; Liao, Y.; Lin, C.; Jia, Y.; Li, Z.; Yang, X. Object Tracking in Satellite Videos Based on Correlation Filter with Multi-Feature Fusion and Motion Trajectory Compensation. *Remote Sens.* **2022**, *14*, 777. [CrossRef]
41. Du, B.; Cai, S.; Wu, C. Object Tracking in Satellite Videos Based on a Multiframe Optical Flow Tracker. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2019**, *12*, 3043–3055. [CrossRef]
42. Xuan, S.; Li, S.; Han, M.; Wan, X.; Xia, G.S. Object Tracking in Satellite Videos by Improved Correlation Filters With Motion Estimations. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 1074–1086. [CrossRef]
43. Li, Y.; Bian, C.; Chen, H. Object Tracking in Satellite Videos: Correlation Particle Filter Tracking Method With Motion Estimation by Kalman Filter. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–12. [CrossRef]
44. Shao, J.; Du, B.; Wu, C.; Gong, M.; Liu, T. HRSiam: High-Resolution Siamese Network, Towards Space-Borne Satellite Video Tracking. *IEEE Trans. Image Process.* **2021**, *30*, 3056–3068. [CrossRef]
45. Hu, Z.; Yang, D.; Zhang, K.; Chen, Z. Object Tracking in Satellite Videos Based on Convolutional Regression Network With Appearance and Motion Features. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2020**, *13*, 783–793. [CrossRef]
46. Yang, J.; Pan, Z.; Wang, Z.; Lei, B.; Hu, Y. SiamMDM: An Adaptive Fusion Network With Dynamic Template for Real-Time Satellite Video Single Object Tracking. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 1–19. [CrossRef]
47. Chen, Y.; Tang, Y.; Yin, Z.; Han, T.; Zou, B.; Feng, H. Single Object Tracking in Satellite Videos: A Correlation Filter-Based Dual-Flow Tracker. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2022**, *15*, 6687–6698. [CrossRef]
48. Jiang, B.; Luo, R.; Mao, J.; Xiao, T.; Jiang, Y. Acquisition of Localization Confidence for Accurate Object Detection. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 816–832. [CrossRef]
49. Newell, A.; Yang, K.; Deng, J. Stacked Hourglass Networks for Human Pose Estimation. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 483–499. [CrossRef]
50. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141. [CrossRef]
51. Guo, Y.; Yin, Q.; Hu, Q.; Zhang, F.; Xiao, C.; Zhang, Y.; Wang, H.; Dai, C.; Yang, J.; Zhou, Z.; et al. The First Challenge on Moving Object Detection and Tracking in Satellite Videos: Methods and Results. In Proceedings of the 26th International Conference on Pattern Recognition (ICPR), Montreal, QC, Canada, 21–25 August 2022; pp. 4981–4988. [CrossRef]
52. Zhao, M.; Li, S.; Xuan, S.; Kou, L.; Gong, S.; Zhou, Z. SatSOT: A Benchmark Dataset for Satellite Video Single Object Tracking. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–11. [CrossRef]
53. Wu, Y.; Lim, J.; Yang, M.H. Object Tracking Benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1834–1848. [CrossRef]
54. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
55. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255. [CrossRef]
56. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015. [CrossRef]
57. Li, F.; Tian, C.; Zuo, W.; Zhang, L.; Yang, M.H. Learning Spatial-Temporal Regularized Correlation Filters for Visual Tracking. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4904–4913. [CrossRef]
58. Danelljan, M.; Bhat, G.; Khan, F.S.; Felsberg, M. ATOM: Accurate Tracking by Overlap Maximization. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 4655–4664. [CrossRef]