remote sensing

MDPI

Article

RAMC: A Rotation Adaptive Tracker with Motion Constraint for Satellite Video Single-Object Tracking

Yuzeng Chen [1], Yuqi Tang [1,2,*], Te Han [1], Yuwei Zhang [1], Bin Zou [1,2] and Huihui Feng [1,2]

[1] School of Geosciences and Info-Physics, Central South University, Changsha 410083, China; yuzengchen@csu.edu.cn (Y.C.); tehanrs@csu.edu.cn (T.H.); zywszsda@163.com (Y.Z.); 210010@csu.edu.cn (B.Z.); hhfeng@csu.edu.cn (H.F.)

[2] Key Laboratory of Metallogenic Prediction of Nonferrous Metals and Geological Environment Monitoring (Central South University), Ministry of Education, Changsha 410083, China

* Correspondence: yqtang@csu.edu.cn

Abstract: Single-object tracking (SOT) in satellite videos (SVs) is a promising and challenging task in the remote sensing community. In terms of the object itself and the tracking algorithm, the rotation of small-sized objects and tracking drift are common problems due to the nadir view coupled with a complex background. This article proposes a novel rotation adaptive tracker with motion constraint (RAMC) to explore how the hybridization of angle and motion information can be utilized to boost SV object tracking from two branches: rotation and translation. We decouple the rotation and translation motion patterns. The rotation phenomenon is decomposed into the translation solution to achieve adaptive rotation estimation in the rotation branch. In the translation branch, the appearance and motion information are synergized to enhance the object representations and address the tracking drift issue. Moreover, an internal shrinkage (IS) strategy is proposed to optimize the evaluation process of trackers. Extensive experiments on space-born SV datasets captured from the Jilin-1 satellite constellation and International Space Station (ISS) are conducted. The results demonstrate the superiority of the proposed method over other algorithms. With an area under the curve (AUC) of 0.785 and 0.946 in the success and precision plots, respectively, the proposed RAMC achieves optimal performance while running at real-time speed.

Keywords: rotation adaptive; optical flow remapping; satellite video; single-object tracking

Citation: Chen, Y.; Tang, Y.; Han, T.; Zhang, Y.; Zou, B.; Feng, H. RAMC: A Rotation Adaptive Tracker with Motion Constraint for Satellite Video Single-Object Tracking. *Remote Sens.* **2022**, *14*, 3108. https://doi.org/10.3390/rs14133108

Academic Editors: Danilo Avola, Alessio Fagioli and Daniele Pannone

Received: 18 May 2022
Accepted: 24 June 2022
Published: 28 June 2022

1. Introduction

Single-object tracking (SOT), a fundamental but challenging task, allows the establishment of object correspondences in a video [1]. It is applied in diverse scenarios, such as surveillance, human-computer interaction, and augmented reality [2,3]. Given only the initial state of an arbitrary object, the tracker aims to estimate its subsequent states in a video [4]. Many studies, including deep-learning-based [5–9] and correlation-filter-based [10–13], have been conducted to improve the tracking effects. Due to the achievements of the convolutional neural network (CNN), researchers have introduced CNNs for object tracking. The CNN-SVM [14] combines CNN with a support vector machine (SVM) [15] to achieve tracking. TCNN [16] and MDNet [6] have demonstrated their performance in object tracking. There are also many trackers based on the Siamese network, such as SiamRPN [7], SiamRPN++ [8] and SiamMask [17]. Deep-SRDCF [18], C-COT [19] and ECO [20] employ deep features extracted from CNNs to enhance the object representations but at the cost of high computational complexity. Correlation filter-based methods have emerged since MOSSE [21] was first proposed. Such methods train the filter by minimizing the output sum of squared errors. The CSK [22] tracker improves upon MOSSE by introducing the circulant matrix and kernel trick. However, CSK still uses a simple raw pixel feature despite improving accuracy and speed. The kernelized correlation filter (KCF) [10] extends CSK by incorporating a multichannel histogram of oriented gradients (HOG) [23] feature and different kernel functions, and it has shown outstanding performance in tracking objects without rotation. Only a few correlation filters [24–26] have considered rotation. In addition, tracking drift is a drawback of correlation filters, which may cause the sample to drift away from the object. Some algorithms (e.g., SRDCF [27] and CSR-DCF [13]) have been proposed to prevent tracking drift at the expense of high time consumption.

Remote sensing observation capabilities have broadened from static images to dynamic videos. In 2013, the SkySat-1 video satellite captured panchromatic videos with a ground sample distance (GSD) of 1.1 m and a frame rate of 30 frames per second (FPS) [28]. In 2016, the International Space Station (ISS) released an ultra-high-definition RGB video with a GSD of 1.0 m and a frame rate of 3 FPS. From 2015 to the present, members of the Jilin-1 satellite constellation produced by China Changchun Satellite Technology Co., Ltd. (Changchun, China) have been launched. Currently, Jilin-1 can capture 30 FPS RGB videos with a GSD of 0.92 m. Video satellites in orbit deliver rich, dynamic information on the Earth's surface and have been successfully used for SOT [29–31], traffic analysis [32], stereo mapping [33] and river velocity measurement [34]. However, compared with natural video (NV), SOT in satellite video (SV) involves many challenges and can be defined as an emerging subject [35]. The main difficulties are two-fold.

(1) The nadir view makes the rotation (in-plane) of the object a common phenomenon, as shown in Figure 1b–d; rotation can induce non-rigid deformation of the object and change the object spatial layout, affecting the performance of the tracking algorithms [36].

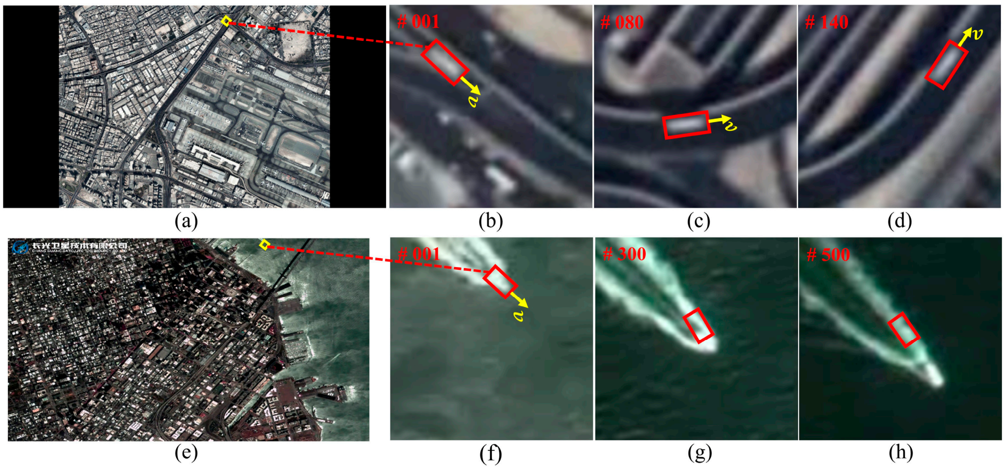


Figure 1. Visualization of object rotation and tracking drift in SVs. The symbol # represents the prefix of the frame number. The current frame is shown in the upper-left corner of each image. The yellow arrow indicates the orientation of the object. (**a**) and (**e**) show the original frames, and the selected objects are enlarged. (**b**–**d**) show the object rotation. (**f**–**h**) show the tracking drift due to the complex background and low contrast of the ship and wake.

(2) The complex background and low contrast between small-sized objects and the background can lead to tracking drift of the algorithms [37], as shown in Figure 1f–h.

The KCF [10] has shown promising performance for SOT in SVs [36–39]. However, HOG-based KCF inherently cannot handle object rotation [40]. The axis-aligned bounding box of the KCF contains more background information, which may cause tracking drift [36,39]. Moreover, compared to the rotating bounding box, it cannot express accurate semantic information, such as the real size and orientation of the object, as shown in Figure 2. To address the object rotation and tracking drift issues of SOT in SVs, we proposed a rotation adaptive tracker with motion constraint (RAMC) consisting of a rotation and a translation branch in this paper. We performed quantitative and qualitative experiments on the space-born SV datasets. The experimental results demonstrate that the RAMC tracker outperforms state-of-the-art algorithms and runs at over 40 FPS. The major contributions are summarized as follows:

(1) We analyze the relationship between the intuitive rotation and the potential translation. And the rotation and translation motion patterns are decoupled by decomposing the rotation phenomenon into a translation solution. It could achieve adaptive rotation estimation when applied to SOT in SVs.

(2) The appearance and motion information, contained in adjacent frames, are then synergized into the framework. It constructs the motion constraint term on the appearance model to prevent tracking drift and guarantee precise localization.

(3) An internal shrinkage strategy is proposed to narrow the gap between the rotating and axis-aligned bounding boxes in the evaluation benchmark. It models axis-aligned rectangles with ellipse-like distributions to optimize the evaluation process.

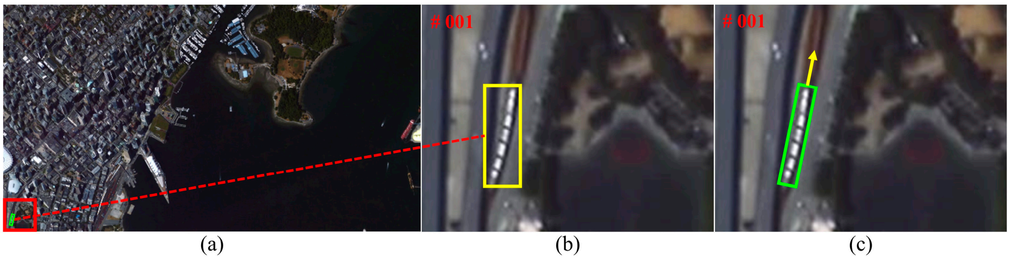


Figure 2. Visualization of different kinds of bounding boxes. The red area in the original frame (**a**) is enlarged; (**b**) shows an axis-aligned bounding box marked in yellow, whereas (**c**) shows a rotating bounding box marked in green.

2. Background

2.1. Satellite Video Single-Object Tracking

As mentioned previously, the main challenges of SOT in SVs are object rotation and tracking drift. A few methods have been proposed to solve the object rotation problem in SVs. Guo et al. [37] detect the orientations of objects by using slope information and output rotating bounding boxes. Xuan et al. [40] rotate the extracted patch with a fixed-angle pool to deal with the object rotation issue and obtained axis-aligned bounding boxes. These methods may be numb to a slight rotational issue. To address tracking drift, some approaches [29–31,36,37,41,42] have built motion models based on the relatively stable motion patterns of objects in SVs. In [30,36,37,41,42], the authors use the properties of the Kalman filter [43] to predict the object position at low tracking confidence, which attenuates the tracking drift. In [29,31], the motion smoothness and centroid inertia models are embedded into the tracking framework to reduce tracking drift. However, most of them [29,31,36,37,41,42] place high demands on positioning accuracy during the initial stage. Other methods [30,38,39,44] extract the motion features contained in adjacent frames to prevent tracking drift. Du et al. [44] combine the three-frame-difference approach and the KCF tracker to obtain the object's position. Shao et al. [30] construct a refining branch modeled on Gaussian mixture models (GMM) to reduce the risk of drifting. In [37,38], the authors use the Lucas–Kanade sparse optical flow [45] feature for SOT in SVs. However, they ignore the directional information of the optical flow. And the sparse optical flow makes it difficult to represent pixel-level motion information [46].

To achieve precise angle estimation and localization of SOT in SVs, this study designs a rotation-adaptive tracking framework with motion constraint. It decouples the rotation and translation motion patterns by decomposing the rotation issue into a translation solution. In addition, it further synergizes the appearance and motion information to enhance the localization performance of SOT in SVs. It guarantees that the proposed method can estimate slight angle differences of objects and prevent tracking drift.

2.2. Kernelized Correlation Filter

KCF [10] has shown promising performance for SOT in SVs [36–39]; we exploit and improve it to address object rotation and tracking drift for accurate semantic representations. This section introduces the KCF [10] framework based on the training and detection processes. It applies dense cyclic samples to explore the structural information of an object. A circulant matrix is used to model specific structures, and the correlation is transformed into element-wise products by a fast Fourier transform (FFT).

Let an $M \times N$ patch x denote a base sample that is centered on an object and is more than twice the size of the object. All cyclic shifts $\{x_{m,n}\}$, $(m,n) \in \{0, \ldots, M-1\} \times \{0, \ldots, N-1\}$ are considered dense sampling over the base sample. They are labeled by a Gaussian function y so that $y(m,n)$ is the label of $x_{m,n}$.

In the training process, the solution ω is obtained by minimizing the ridge regression error [10], as follows:

$$\min_w \sum_{m,n} |\langle \varphi(x_{m,n}), \omega \rangle - y(m,n)|^2 + \lambda \| \omega \|^2 \tag{1}$$

where φ is the Hilbert space mapping induced by kernel κ. The inner product is defined as $\varphi(x), \varphi(g) = \kappa(x,g)$. A constant $\lambda \geq 0$ is a regularization term that avoids overfitting. After a powerful nonlinear regression using the kernel trick, the solution ω is

$$\omega = \alpha(m,n)\varphi(x_{m,n}). \tag{2}$$

The discrete Fourier transform (DFT) of a vector is denoted by a hat (ˆ). The kernel matrix is a circulant matrix in commonly used kernel functions [10]. Thus, the dual-space coefficient α is

$$\hat{\alpha} = \frac{\hat{y}}{\hat{k}^{xx} + \lambda}, \tag{3}$$

where $k^{xx} = \kappa(x_{m,n}, x)$. A Gaussian kernel is employed to compute the kernel correlation k^{xx} with element-wise products in the frequency domain. For a patch with C feature channels, the base sample is $x = [x_1, x_2, \ldots, x_C]$. Therefore, we have

$$k^{xx'} = \exp\left(-\frac{1}{\sigma^2}\left(\| x \|^2 + \| x' \|^2 - 2\mathcal{F}^{-1}\left(\sum_{i=1}^{C} \hat{x}_i \odot \hat{x}_i'^{*}\right)\right)\right) \tag{4}$$

where $\mathcal{F}^{-1}$ is the inverse Fourier transform (IFT), $\odot$ denotes element-wise products, $*$ denotes the complex conjugate, and i is the index of feature channels.

In the detection process, patch z in a new frame equal to the size of x is cropped out in the center of the object in the previous frame. The response map $f(z)$ is solved by:

$$f(z) = \left(\mathcal{F}^{-1}\left(\hat{k}^{xz} \odot \hat{\alpha}\right)\right). \tag{5}$$

The object position is then obtained by determining the maximum value of $f(z)$. To adapt to changes in the object, the two coefficients x and α are updated [10].

3. Methodology

Figure 3 shows the overall framework of the proposed method, including the rotation and translation branches. The rotation branch calculates the rotation angle $\alpha(t)$ of the object to provide an accurate orientation representation. Moreover, the angle is adopted for the translation branch to yield a stable response map. The translation branch synergizes the appearance and motion information contained in adjacent frames for positioning to prevent tracking drift. These two complementary branches are unified for the SOT in SVs.

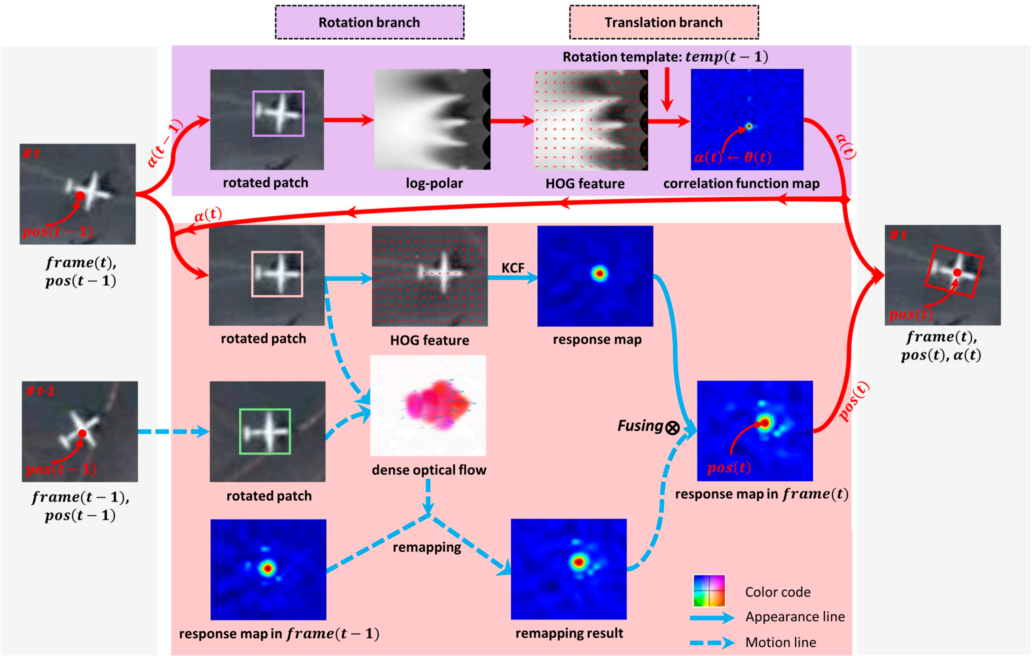


Figure 3. Overall framework of the proposed RAMC algorithm. $pos(t-1)$ and $\alpha(t-1)$ denote the position and angle of the object at frame $t-1$, respectively. The color code is shown for visualizing the optical flow, in which the color denotes the displacement direction, and the saturation represents the displacement magnitude. The appearance and motion lines represent the positioning process of appearance and motion information, respectively. Two frames separated by T are selected as adjacent frames to show the overall tracking framework. It yields more intuitive rotation and translation of the object for visualization.

3.1. Rotation Branch for Adaptive Angle Estimation

Object rotation is common in SVs because of the nadir view. It can cause the spatial layout between the object and background to change, challenging the accuracy and robustness of object tracking. Since the log-polar conversion can convert the rotation problem into a potential translation problem, we propose an adaptive angle-estimation method incorporating Fourier–Mellin registration [47]. The rotation branch consists of three main parts: log-polar conversion, feature extraction, and phase correlation, which are introduced briefly.

For the log-polar conversion, $I(u,v)$ and $I'(\rho,\ \theta)$ denote the patches in Cartesian coordinates and log-polar coordinates, respectively. After giving the pivot point $(u_0,\ v_0)$ and the reference axis (u axis), it is obtained that:

$$\rho = \lg\left(\sqrt{(u-u_0)^2 + (v-v_0)^2}\right), \tag{6}$$

$$\theta = \arctan\left(\frac{v-v_0}{u-u_0}\right), \tag{7}$$

where ρ is the log distance between the original point (u,v), the pivot point $(u_0,\ v_0)$, lg indicates log_{10}, and θ denotes the angle between the reference axis and the line through the pivot and original point.

$I_2(u,v)$ is a rotated replica of $I_1(u,v)$ with rotation θ_0. The correspondences in Cartesian coordinates are

$$I_2(u,v) = I_1(u\cos\theta_0 + v\sin\theta_0, -u\sin\theta_0 + v\cos\theta_0), \tag{8}$$

In log-polar coordinates, their correspondences are

$$I_2{}'(\rho,\ \theta) = I_1{}'(\rho,\ \theta - \theta_0). \tag{9}$$

It can be seen that the rotation between I_2 and I_1 is deduced as translation between $I_2{}'$ and $I_1{}'$. By calculating the offset θ_0, the angle difference between the two patches can be obtained.

For the feature extraction, the HOG feature is sensitive to the rotation of the object and can be applied to discriminate the angle difference [40]. In this study, the HOG feature is extracted in log-polar coordinates and used to calculate the offset θ_0 by phase correlation.

For the phase correlation, it can be used to match images translated to each other [47]. Therefore, it is employed to solve θ_0. $f_1(u,\ v)$ and $f_2(u, v)$ denote the two 2D patches. $f_2(u, v)$ has a $(u_0,\ v_0)$ displacement from $f_1(u,\ v)$ along the u axis and v axis, as follows:

$$f_2(u,\ v) = f_1(u - u_0,\ v - v_0). \tag{10}$$

Their Fourier transforms are related by

$$F_2(\alpha,\ \beta) = e^{-j2\pi(\alpha u_0 + \beta v_0)} F_1(\alpha,\ \beta), \tag{11}$$

where $F_1(\alpha,\ \beta)$ and $F_2(\alpha,\ \beta)$ denote the DFT of $f_1(u,\ v)$ and $f_2(u,\ v)$, respectively. The cross-phase spectra of $F_1(\alpha,\ \beta)$ and $F_2(\alpha,\ \beta)$ are defined as:

$$\widetilde{C}(\alpha,\ \beta) = \frac{F_1(\alpha,\ \beta) F_2^*(\alpha,\ \beta)}{\left|F_1(\alpha,\ \beta) F_2^*(\alpha,\ \beta)\right|} = e^{j2\pi(\alpha u_0 + \beta v_0)}, \tag{12}$$

where $e^{j2\pi(\alpha u_0 + \beta v_0)}$ is the u and v axis translation in the Fourier domain. By applying the 2D IFT to $\widetilde{C}(\alpha,\ \beta)$, the phase correlation function φ of the spatial domain can be obtained by

$$\varphi(u - u_0, v - v_0) = \mathcal{F}^{-1}\left(e^{j2\pi(\alpha u_0 + \beta v_0)}\right). \tag{13}$$

In the φ, the location $(u_0,\ v_0)$ corresponds to the offset between the two images and can be computed for

$$(u_0,\ v_0) = argmax(\varphi(u - u_0, v - v_0)). \tag{14}$$

Through phase correlation, we can finally obtain the offset $\theta_0 = v_0$, which is the angle difference between $I_2(u, v)$ and $I_1(u, v)$.

In tracking, the rotation template $temp(1)$ for the first frame is obtained by extracting the HOG feature $z_{hog}(1)$ of the log-polar patch representation. In the subsequent frames, the extracted result $z_{hog}(t)$ at frame t is utilized to compute the angle difference $\theta(t)$ between the $temp(t-1)$ and $z_{hog}(t)$, as shown in Figure 3. Moreover, to adapt to object changes (e.g., illumination and deformation), the rotation template $temp(t)$ at frame t is updated by the learning rate η_θ. The pseudocode of rotation branch procedure is shown in Algorithm 1. Compared with the methods [37,40] that use slope and angle pool to estimate the rotation angle of the object, the proposed method may be suitable for achieving accurate angle estimation and orienting the bounding boxes to a real state in SVs. Stable response maps can also be obtained under object rotation, thereby enhancing the positioning of the translation branch.

Algorithm 1 Rotation Branch Procedure

```
Input: frame index: $t$, total frames: $N$, image frame: $frame(t)$, position: $pos(t)$, rotation template: $temp(t)$, learning rate of rotation templates: $\eta_\theta$;
Output: rotation angle of the object at $frame(t)$: $\alpha(t)$;
for $t = 1; t <= N; t++$
    if $t = 1$
        /* carry out initialization in the first frame */
        /* set a tracked object by $pos(1)$ */
        Initialize the angle $\alpha(t)$ in the first frame;
        Initialize the rotation template $temp(1)$ in the first frame;
    else
        Extract rotated patch $z_\alpha(t)$ at $pos(t-1)$ at angle $\alpha(t-1)$ from $frame(t)$;
        Convert $z_\alpha(t)$ to log-polar coordinates and obtain patch $z_{lp}(t)$;
        Extract HOG feature $z_{hog}(t)$ of patch $z_{lp}(t)$;
        Calculate the phase correlation function $\varphi$ between $z_{hog}(t)$ and $temp(t-1)$;
        Estimate angle difference $\theta(\mathrm{t}) \leftarrow argmax(\varphi)$;
        Update angle $\alpha(t) \leftarrow \alpha(t-1) + \theta(t)$;
        Update template $temp(t) \leftarrow (1-\eta_\theta) * temp(t-1) + \eta_\theta * z_{hog}(t)$;
      return: $\alpha(t)$
      end if
end for
```

3.2. Translation Branch with Motion Constraint

The object angle $\alpha(t)$ in frame t can be estimated using the rotation branch. The next stage is to determine the object position (x, y) on the basis of the $\alpha(t)$. In the translation branch, the input patch of frame t is first rotated by $\alpha(t)$, and then the rotated patch is fed into the KCF [10] and optical flow remapping (OFR) modules for accurate positioning. In the KCF module, the issue caused by the object rotation is removed to ensure that tracking can be achieved using the appearance information of the object, as described in Section 2.2. In the OFR module, the optical flow represents the apparent motion of the brightness patterns and captures the motion magnitude and direction information between adjacent frames [48]. Therefore, the motion state of the object in the previous frame can be remapped to the current frame using the optical flow feature. To achieve the per-pixel motion constraint information, we employ Farneback dense optical flow [49] to remap previous response maps into the current frame.

For the dense optical flow, it approximates the neighborhoods of each pixel using a quadratic polynomial. Given a local signal model, the local coordinate system $E(X)$ of pixel $X = (x, y)^T$ can be expressed as

$$E(X) = X^T A X + B^T X + C, \tag{15}$$

where $A \in \mathbb{R}^{2\times 2}$, $B \in \mathbb{R}^2$ and C denote the coefficients of the quadratic polynomial, which are estimated using a weighted least-squares method. The polynomial coefficients of the neighborhood change with the pixels of the frame. When a pixel is moved by displacement D, a new local system is constructed as follows:

$$\begin{aligned} E_2(X) = E_1(X-D) &= (X-D)^T A_1 (X-D) + B_1{}^T (X-D) + C_1 \\ &= X^T A_1 X + (B_1 - 2A_1 D)^T X + D^T A_1 D - B_1{}^T D + C_1 \\ &= X^T A_2 X + B_2{}^T X + C_2 \end{aligned}, \tag{16}$$

where $A_2 = A_1$, $B_2 = B_1 - 2A_1 D$ and $C_2 = D^T A_1 D - B_1{}^T D + C_1$ according to the equality of the polynomial coefficients. From the equation $B_2 = B_1 - 2A_1 D$, we obtain $D = -\frac{1}{2} A_1{}^{-1}(B_2 - B_1)$ and solve it by minimizing the objective function:

$$e(X) = \| \, AD - \Delta B \, \|^2, \tag{17}$$

where $\Delta B = -\frac{1}{2}(B_2 - B_1)$. In order to suppress the excessive noise caused by single-point optimization [49], the neighborhood δ of pixel X is integrated to obtain the solution from

$$argmin_{D(X)} \sum_{\Delta X \in \delta} \omega(\Delta X) \| \ A(X + \Delta X)D(X) - \Delta B(X + \Delta X) \ \|^2, \tag{18}$$

where $\omega(\Delta X)$ denotes the 2D Gaussian weight function of the neighborhood points. The displacements of the pixels can be solved by

$$D(X) = \left(\sum \omega A^T A\right)^{-1} \sum \omega A^T \Delta B. \tag{19}$$

The optical flow obtained represents the direction and magnitude of each pixel in the frame. The response map containing the historical states of the object is then remapped to the current frame for positioning.

For the remapping, it is the process of transferring each pixel of the original image $g(x, y)$ with size $M \times N$ to the target image $G(x, y)$. The process is

$$G(x_i, y_j) = g(R(x_i, y_j)), i \in \{1, \cdots, M\}, \ j \in \{1, \cdots, N\}, \tag{20}$$

where R denotes the mapping relationship that specifies the motion direction and magnitude of each pixel in the original image. Therefore, the optical flow at frame t can be regarded as the mapping relationship $R(x_i, y_j) = D(x_i, y_j)$, and the previous response map at frame $t - 1$ is remapped to the current frame t, as shown in Figure 4.

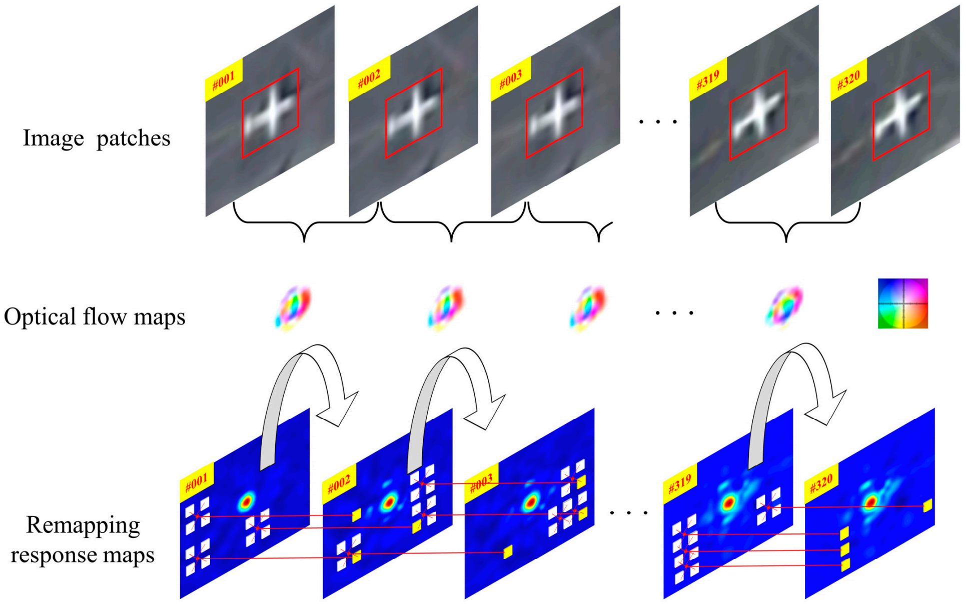


Figure 4. Visualization of the optical flow remapping process. In the last row, each pixel of the target response map is traversed to calculate its corresponding position in the previous response map. If the position does not exist, the values of its neighboring pixels will be interpolated to determine the pixel value in the target response map.

We implement object tracking with motion constraint in the SVs in the translation branch. To comprehensively exploit the motion and appearance information, the OFR module works with the KCF module to obtain the object position $pos\ (t)$, thereby addressing the tracking drift.

4. Experimental Details and Analysis

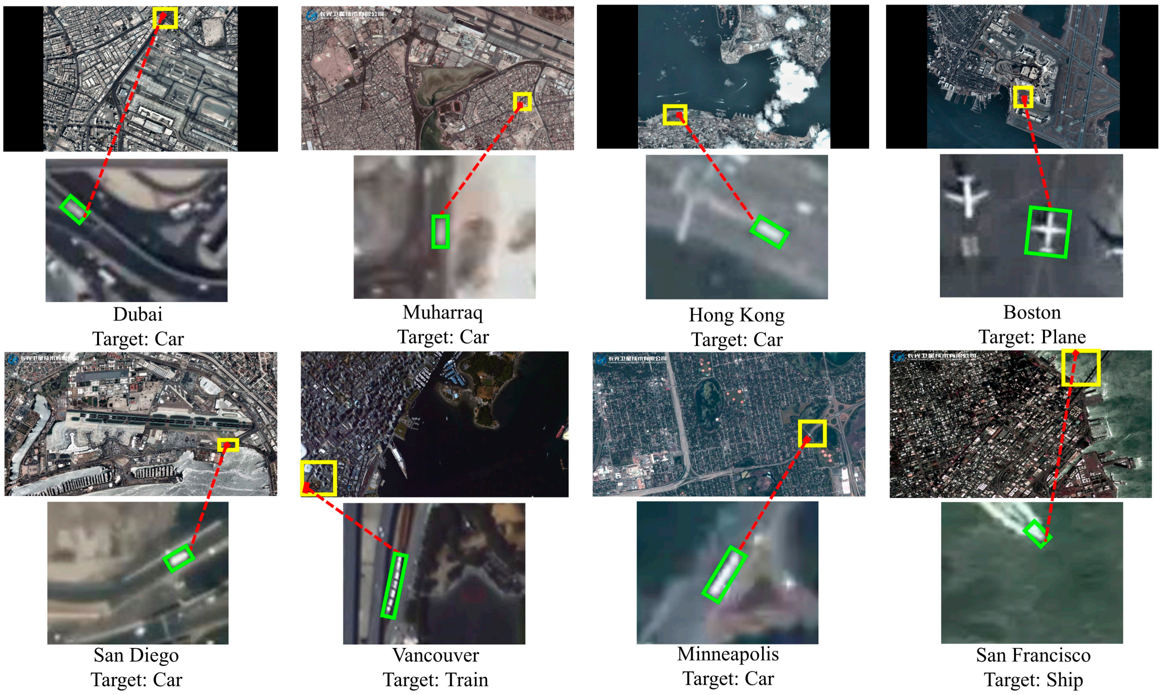
4.1. Experimental Settings

4.1.1. Datasets

We conduct extensive experiments using eight high-resolution SVs. Seven SVs are obtained from the Jilin-1 satellite constellation launched by Chang Guang Satellite Technology Co., Ltd. (Changchun, China). Moreover, only the Vancouver dataset is acquired using a high-resolution iris camera installed on the ISS. These datasets are divided into two groups: those with rotation (i.e., Dubai, Muharraq, Hong Kong, and Boston) and those without rotation (i.e., San Diego, Vancouver, Minneapolis, and San Francisco) to demonstrate the tracking effectiveness of the proposed method for rotating and non-rotating objects. Tracked targets include cars, planes, trains, and ships. These objects are represented by rotating bounding boxes annotated with four corner coordinates. One region of each SV is cropped for clear visualization. Table 1 provides detailed information about the datasets. Figure 5 shows the first frames, cropped regions, and tracked objects.

Table 1. Details of the SV datasets. "px" = pixels.

SVs	Frame Size (px)	Cropped Region Size (px)	GSD (m)	Frame Number	Target	Target Size (px)	Rotation
Dubai	4096 × 3072	256 × 218	0.92	147	Car	15 × 7	√
Muharraq	4096 × 2160	174 × 193	0.92	500	Car	13 × 6	√
Hong Kong	4096 × 3720	268 × 216	0.92	164	Car	14 × 7	√
Boston	4096 × 3720	215 × 300	0.92	320	Plane	36 × 33	√
San Diego	4090 × 2160	135 × 94	0.92	228	Car	13 × 6	–
Vancouver	3840 × 2160	501 × 487	0.92	405	Train	16 × 80	–
Minneapolis	4090 × 2160	336 × 328	0.92	268	Car	24 × 7	–
San Francisco	3840 × 2160	501 × 501	1.00	500	Ship	15 × 9	–



Dubai
Target: Car

Muharraq
Target: Car

Hong Kong
Target: Car

Boston
Target: Plane

San Diego
Target: Car

Vancouver
Target: Train

Minneapolis
Target: Car

San Francisco
Target: Ship

Figure 5. The SV datasets used in the experiments. A region marked by a yellow rectangle in each dataset is cropped out, and the tracked object, marked by the green rectangle, is displayed enlarged.

4.1.2. Evaluation Metrics

To measure the performance of the tracking algorithms, two protocols (success and precision plots) in the online tracking benchmark [50,51] are used. The success plot displays the percentages of scenarios in which the *overlap* between the estimated bounding box b_e and ground truth b_g is larger than the threshold of $t_s \in [0,\ 1]$:

$$overlap = \frac{|b_g \cap\ b_e|}{|b_g \cup\ b_e|}, \tag{21}$$

where $\cap$ and $\cup$ denote intersection and union operators, respectively, and $|\cdot|$ denotes the number of pixels in the region. The precision plot records the percentage of scenarios in which the center location error (CLE) between the estimated location and ground truth is smaller than the threshold $t_p \in [1,\ 50]$. The area under the curve (AUC) of the success and precision plots is selected to rank all trackers, avoiding unfair comparisons due to specific thresholds. We mainly rank trackers based on the AUC of the success plot because of its representativeness in evaluation [52]. The FPS is used to evaluate tracking speed.

To narrow the gap between the rotating and axis-aligned bounding boxes in the evaluation, we propose an internal-shrinkage (IS) strategy, as shown in Figure 6.

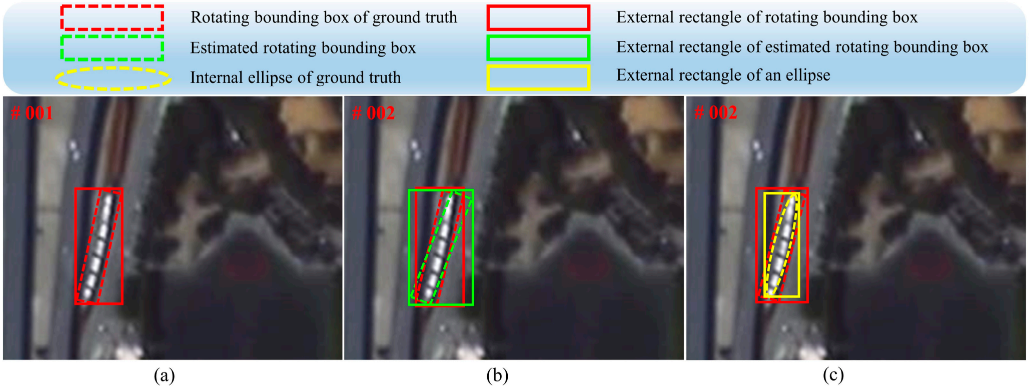

Rotating bounding box of ground truth
Estimated rotating bounding box
Internal ellipse of ground truth
External rectangle of rotating bounding box
External rectangle of estimated rotating bounding box
External rectangle of an ellipse
# 001
# 002
# 002
(a)
(b)
(c)

Figure 6. Visualization of different bounding boxes. (**a**) shows the external rectangle of the rotating bounding box. (**b**) shows the effect of angle deviation on the external rectangle. (**c**) presents the internal-shrinkage strategy for evaluation.

Most algorithms receive (initialize) and yield (output) axis-aligned bounding boxes. To optimize the evaluation process, we attempt to initialize and evaluate the algorithms using an external rectangle, as shown in Figure 6a. In this way, the initial bounding box contains many backgrounds, and even a small angle deviation may greatly affect the overlap between the estimated external rectangle and the ground truth rectangle, as shown in Figure 6b. Considering that the object of SV appears as an ellipse-like distribution pattern, we compute the internal ellipse followed by its external rectangle, as shown in Figure 6c. Finally, we use the internal rectangle to initialize trackers that can only receive axis-aligned labels. In this way, the estimated rotating bounding boxes are converted to an external rectangle using the proposed IS strategy for the evaluation.

4.1.3. Implementation Details

Considering the relatively slight object changes (e.g., illumination, deformation) in a short time, the learning rate of the rotation template is set as $\eta_\theta = 0.01$. The cell size and orientation of the HOG feature are set to 4×4 and 9, respectively, for accurate tracking, as commonly used in [36,40]. The other parameters related to KCF are referred to [10]. The trackers are executed on a workstation with a 3.20 GHz Intel(R) Xeon(R) Gold 6134 CPU (32-core) and NVIDIA GeForce RTX 2080 Ti GPU.

To correctly select the fusion weight ω for optical flow remapping, we randomly selected three of the eight SVs for the experiments. Table 2 presents the experimental details, and Figure 7 shows the AUC of the success and precision plots. It can be seen that both the AUC of the success and precision plots tended to increase and then decrease as the weights increased. The results are ideal when the weight is approximately 0.36. Therefore, this weight $\omega = 0.36$ is used in subsequent experiments.

Table 2. AUC of the success and precision plots on a randomly picked data set.

Weights	0	0.06	0.12	0.18	0.24	0.30	0.36	0.42	0.48	0.54	0.60	0.66	0.72	0.78	0.84	0.90	0.96
Success plot	0.473	0.642	0.663	0.675	0.611	0.476	0.729	0.720	0.701	0.699	0.681	0.270	0.277	0.106	0.081	0.085	0.192
Precision plot	0.579	0.765	0.805	0.831	0.764	0.597	0.881	0.871	0.863	0.861	0.846	0.364	0.355	0.152	0.108	0.153	0.283

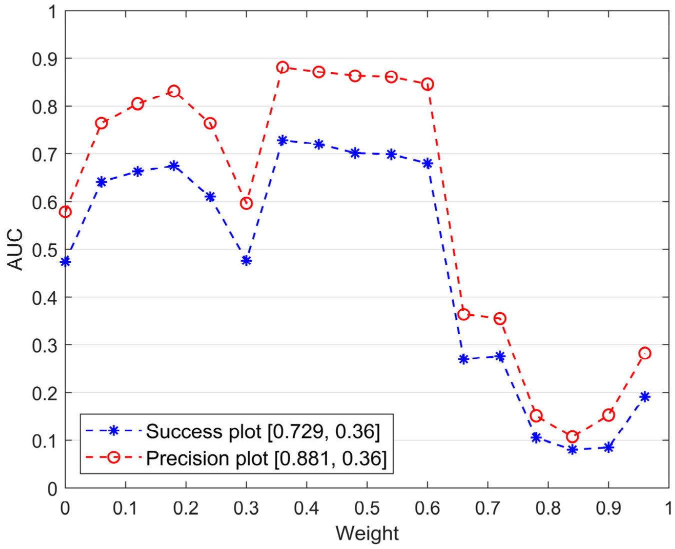


Figure 7. Effects of the weights of optical flow remapping on tracking performance. The legend presents the maximum values of the AUC and their indexes.

4.2. Ablation Experiments

The proposed method incorporated two major improvements: (1) angle estimation (AE) and (2) optical flow remapping (OFR). To validate their performance, three variants of RAMC are tested: Baseline (only KCF [10]), Base_AE (add AE to Baseline), and Base_OFR (add OFR to Baseline). Table 3 lists the components of these variants and the experimental results, and Figure 8 shows the success and precision plots. By comparing the Baseline and Base_AE, Base_AE yields a 13.9% and 18% gain in the AUC of the success and precision plots, respectively, after adding the AE. When comparing the proposed RAMC and Base_OFR, we find a 14.5% and 3.6% reduction in the AUC of the success and precision plots, respectively, when removing the AE. Owing to the absence of AE, object rotation can adversely affect the tracking performance. By comparing the Baseline and Base_OFR, Base_OFR obtains 34.4% and 34.2% in the AUC of the success and precision plots, respectively, after adding the OFR. In contrast with Base_AE, the proposed RAMC yields a 35% and 19.8% improvement in the AUC of the success and precision plots, respectively, when adding the OFR. This is because OFR exploits the underlying motion information in adjacent frames to prevent tracking drift. Owing to the synergy of the AE for the rotation branch and OFR for the translation branch, the proposed RAMC yields optimal performance.

Table 3. Components and results of ablation experiments.

Trackers	KCF	+AE	+OFR	Success Plot	Precision Plot
Baseline	√	–	–	0.296	0.568
Base_AE	√	√	–	0.435	0.748
Base_OFR	√	–	√	0.640	0.910
RAMC	√	√	√	0.785	0.946

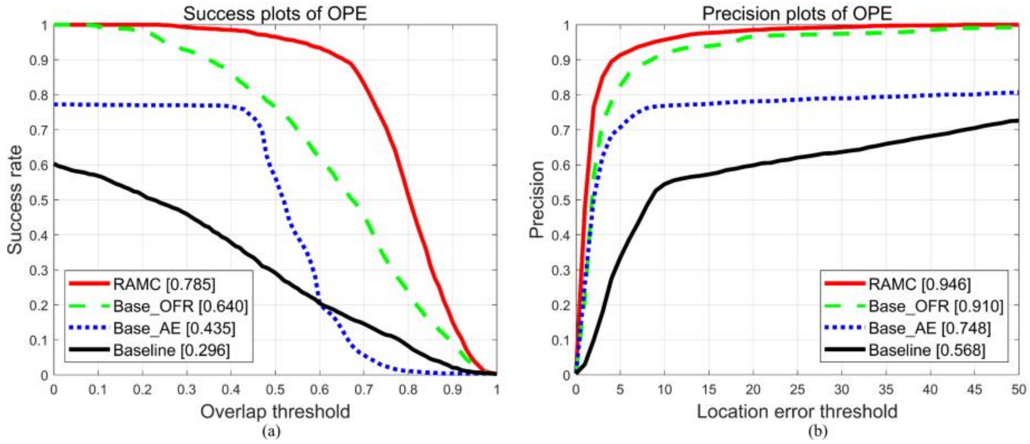


Figure 8. Success plot (**a**) and precision plot (**b**) of the variant trackers on datasets. The values in the legends are the AUC. "OPE" = one-pass evaluation, which initializes a tracker in the first frame and lets it run to the end of the sequence.

4.3. Comparisons on Satellite Video Datasets

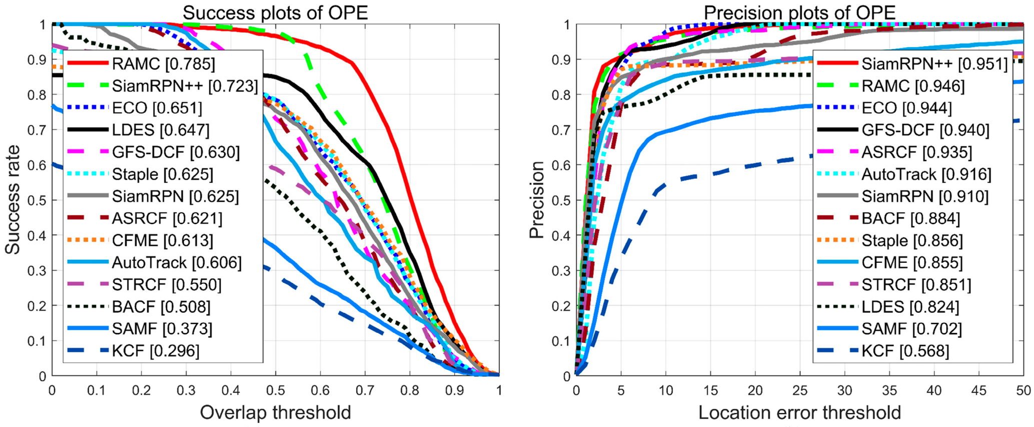
4.3.1. Quantitative Evaluation

We compare the proposed RAMC with 13 competing algorithms: KCF [10], SAMF [53], Staple [11], BACF [12], ECO [20], SiamRPN [7], STRCF [54] SiamRPN++ [8], ASRCF [55], LDES [26], GFS-DCF [56], AutoTrack [57], and CFME [36]. Table 4 summarizes the characteristics and experimental results for these trackers, sorted by the AUC of the success plot. Figure 9 shows the average success and precision plots. The proposed RAMC performs optimally with an AUC of 0.785 in the success plot and 0.946 in the precision plot. Algorithms that cannot cope with object rotation (such as KCF, SAMF, AutoTrack, and CFME) generally achieve inferior performances. The baseline KCF achieves the worst performance owing to the limited representation of the HOG. AutoTrack outperforms SAMF by 23.3% and 21.4% in the success and precision plots, respectively, by exploiting local and global information. CFME obtains AUC of 0.613 and 0.855 in the success and precision plots, respectively, because it uses the motion model to mitigate tracking drift. However, they cannot adapt to object rotation. Generally, algorithms that can cope with rotation but cannot cope with tracking drift (such as SiamRPN, GFS-DCF, LDES, ECO, and SiamRPN++) improve tracking performance. SiamRPN, GFS-DCF, ECO, and SiamRPN++ use rotation-invariant deep features to achieve satisfactory performance. However, these algorithms ignore the motion information hidden in adjacent frames and encounter tracking drift. Compared with the champion ECO of the VOT2017 challenge, RAMC produces a gain of 13.4% and 0.7% in the success and precision plots, respectively. Compared with SiamRPN++, which uses deep networks and a multi-layer aggregation mechanism, RAMC achieves a 6.2% higher success rate owing to the consideration of angle and motion information. The results suggest that RAMC can synergize the AE of the rotation branch and the OFR of the translation branch to cope with object rotation and tracking drift issues, yielding superior tracking effects. Meanwhile, it can run at over 40 FPS. The frame rate of SVs is usually 10 FPS. A tracker with a speed higher than 20 FPS can be considered as a real-time algorithm [38,39].

To further evaluate all the algorithms, we conducted two sets of experiments on datasets with rotation (Rotation) and without rotation (Nrotation). Table 5 summarizes the experimental results, and Figure 10 shows the success and precision plots. For the Rotation dataset, the proposed RAMC obtains better accuracy than that of KCF, SAMF, AutoTrack, and CFME due to considering the rotation issue. The LDES can estimate the rotation angle but ignores the inter-frame motion information. In addition, slight background jitter can affect its accuracy in estimating the angle of small-sized objects. Compared to ECO, ASRCF, and GFS-DCF, which use rotation-invariant deep features, the proposed RAMC exceeds them by 18.9%, 17.5%, and 14% in the success plot, respectively. RAMC has the highest AUC of 0.796 in the success plot, followed by SiamRPN++ (0.732) in second place, and SiamRPN (0.716) in third place. This is because it considers inter-frame motion information on top of the AE, resulting in optimal performance. Compared with SiamRPN, RAMC yields a reduction of only 0.6% in the precision plot. This is because the complex background affects the direction of the optical flow vector, causing motion constraint bias.

Table 4. Details of trackers and experimental results on datasets. The top three of each metric is bolded. "MR" = Mechanisms for Rotation. "MTD" = Mechanisms for Tracking Drift. For trackers, "TGRS" = IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING. For framework, "KCF" = Kernelized Correlation Filter, "DCF" = Discriminative Correlation Filter, "SiameseFC" = Fully Convolutional Siamese Network, "II" = Integral Image and "CCF" = Continuous Convolution Filter. For features, "HOG" = HOG, "CN" = Color Names, "ConvFeat" = Convolutional Features, "CH" = Color Histogram and "OF" = Optical Flow.

Trackers	Framework	Features	MR	MTD	Success Plot	Precision Plot	FPS
KCF (TPAMI 2015)	KCF	HOG	–	–	0.296	0.568	**231.53**
SAMF (ECCV 2014)	KCF	HOG + CN	–	–	0.373	0.702	48.03
BACF (ICCV 2017)	KCF	HOG	–	–	0.508	0.884	52.98
STRCF (CVPR 2018)	DCF	ConvFeat + HOG + CN	√	–	0.550	0.851	18.89
AutoTrack (CVPR 2020)	DCF	HOG + CN	–	–	0.606	0.916	51.90
CFME (TGRS 2020)	KCF	HOG	–	√	0.613	0.855	**104.36**
ASRCF (CVPR 2019)	KCF	ConvFeat + HOG	√	–	0.621	0.935	21.17
SiamRPN (CVPR 2018)	SiameseFC	ConvFeat	√	–	0.625	0.910	62.76
Staple (CVPR 2016)	DCF + II	HOG + CH	√	–	0.625	0.856	**85.32**
GFS-DCF (ICCV 2019)	DCF	ConvFeat + HOG + CN	√	–	0.630	0.940	3.20
LDES (AAAI 2019)	DCF	HOG + CH	√	–	0.647	0.824	21.72
ECO (CVPR 2017)	CCF	ConvFeat	√	–	**0.651**	**0.944**	2.63
SiamRPN++ (CVPR 2019)	SiameseFC	ConvFeat	√	–	**0.723**	**0.951**	33.03
RAMC (proposed)	KCF	HOG + OF	√	√	**0.785**	**0.946**	42.47



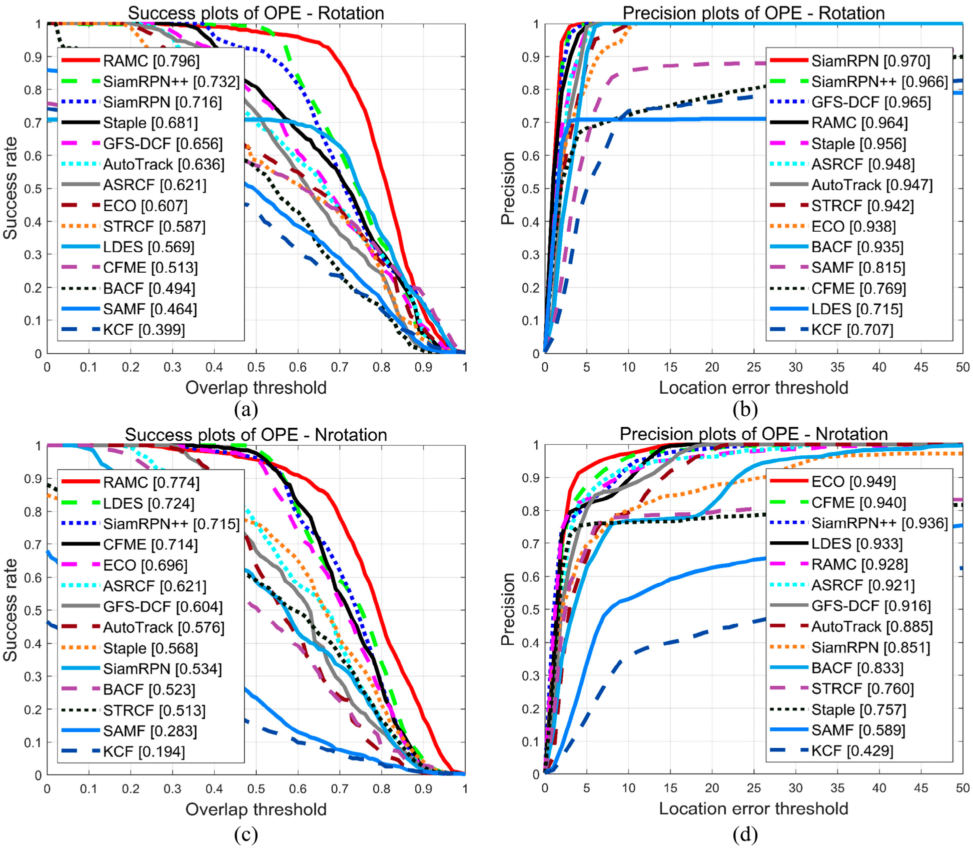
(a) (b)

Figure 9. Success plot (**a**) and precision plot (**b**) of all algorithms.

For the Nrotation dataset, RAMC obtains top-ranked results with AUC of 0.774 and 0.928 in the success and precision plots, respectively. SiamRPN++ produces a satisfactory performance. However, it uses only the appearance information of objects while ignoring motion information. Small-sized objects with similar surroundings may cause tracking drift and degrade tracking effects. Compared to SiamRPN++, RAMC gains the AUC by 5.9% of the success plot. In comparison to ECO, RAMC achieves a 7.8% gain in the success plot. Compared to CFME, RAMC also gains the AUC by 6% of the success plot. The LDES achieves promising performance, with an AUC of 0.724, ranking second in the success plot, since it employs a block coordinate descent (BCD) solver to find the best state for coping with illumination changes and deformations. However, RAMC improves the AUC by 5.0% in the success plot by extracting the motion information contained in adjacent frames. Overall, the proposed RAMC can synergize the AE of the rotation branch and OPR of the translation branch to achieve accurate and robust tracking.

Table 5. The results of all algorithms on the Rotation and Nrotation datasets. The top three of each metric is bolded.

Trackers	Rotation		Nrotation	
	Success Plot	**Precision Plot**	**Success Plot**	**Precision Plot**
KCF (TPAMI 2015)	0.399	0.707	0.194	0.429
SAMF (ECCV 2014)	0.464	0.815	0.283	0.589
BACF (ICCV 2017)	0.494	0.935	0.523	0.833
STRCF (CVPR 2018)	0.587	0.942	0.513	0.760
AutoTrack (CVPR 2020)	0.636	0.947	0.576	0.885
CFME (TGRS 2020)	0.513	0.769	0.714	**0.940**
ASRCF (CVPR 2019)	0.621	0.948	0.621	0.921
SiamRPN (CVPR 2018)	**0.716**	**0.970**	0.534	0.851
Staple (CVPR 2016)	0.681	0.956	0.568	0.757
GFS-DCF (ICCV 2019)	0.656	**0.965**	0.604	0.916
LDES (AAAI 2019)	0.569	0.715	**0.724**	0.933
ECO (CVPR 2017)	0.607	0.938	0.696	**0.949**
SiamRPN++ (CVPR 2019)	**0.732**	**0.966**	**0.715**	**0.936**
RAMC (proposed)	**0.796**	0.964	**0.774**	0.928



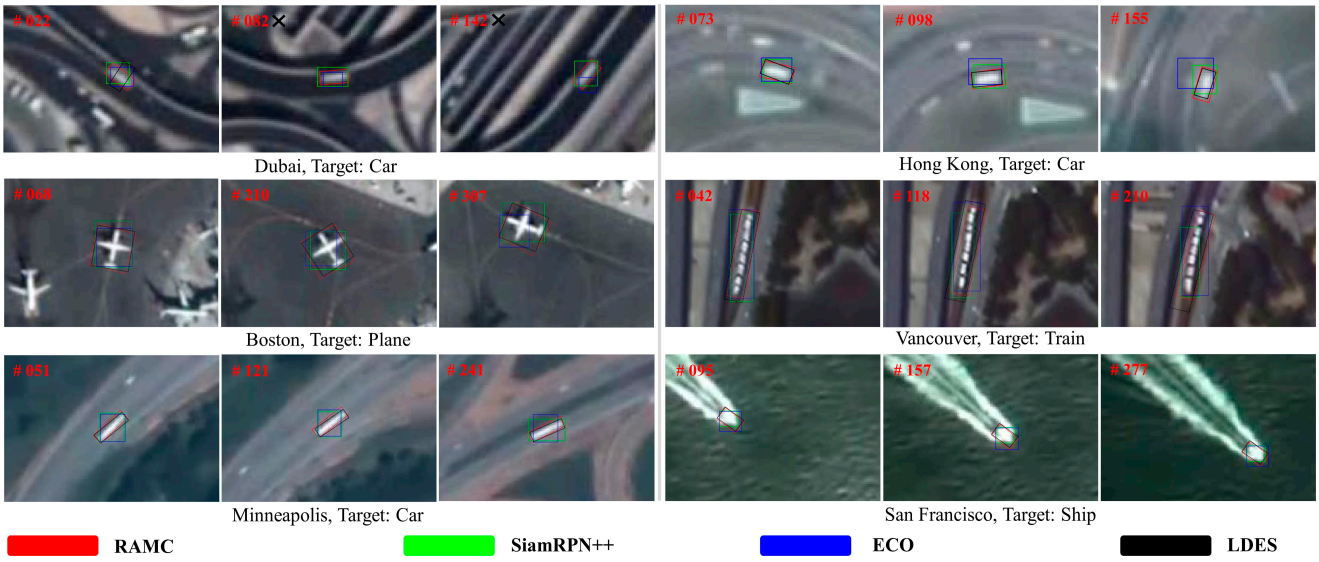
(a) (b)

(c) (d)

Figure 10. Success and precision plots of all algorithms on the Rotation and Nrotation datasets. (**a**) and (**b**) indicate the success plot and precision plot under the Rotation datasets, respectively. (**c**) and (**d**) indicate the success plot and precision plot under the Nrotation datasets, respectively.

4.3.2. Qualitative Evaluation

Figure 11 presents visualization comparisons of the top four trackers. In the Dubai data, only RAMC and SiamRPN++ successfully track the car; however, SiamRPN++ outputs axis-aligned bounding boxes that cannot provide the rotation angle of the object. The ECO fails to deal with scale changes and encounters tracking drift. The LDES cannot adapt to the object's fast rotation and similar background and eventually loses the object. In the Boston data, RAMC, LDES, and SiamRPN++ all locate the plane. Moreover, RAMC and LDES can also obtain the object's angle information. The angle estimation effects of RAMC are superior to those of LDES, owing to the motion constraint. The ECO gradually drifts away from the object. RAMC and ECO accurately locate the train in the Vancouver data. Compared with ECO, RAMC can obtain more semantic representations, including the size and motion direction. The bounding boxes of SiamRPN + + tend to be smaller and drift away from the train. This is because motion blur deteriorates the tracking template of the train, which affects the RPN module migrated from the faster R-CNN [58] to regress to the correct position and scale. LDES incorrectly estimates the scale owing to motion blur. In the other cases shown in Figure 11, RAMC is better at estimating the rotation angle and position of the objects. The quantitative results verify its outstanding performance in estimating the angle and preventing tracking drift.



Dubai, Target: Car

Hong Kong, Target: Car

Boston, Target: Plane

Vancouver, Target: Train

Minneapolis, Target: Car

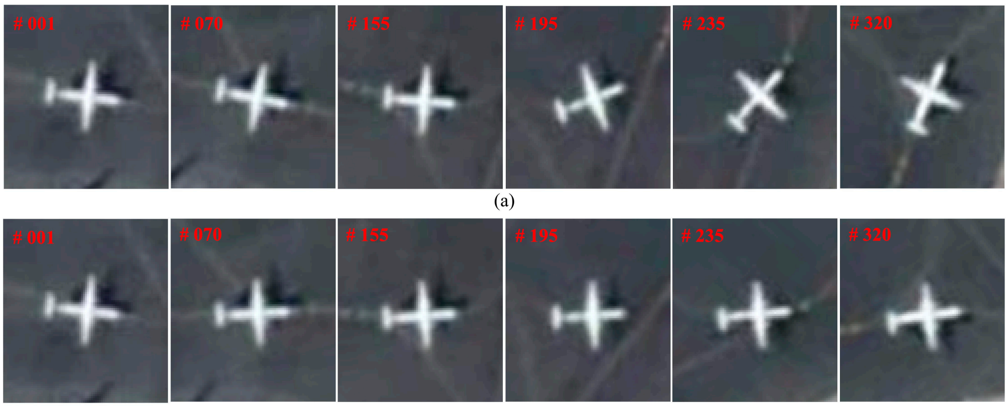
San Francisco, Target: Ship

Figure 11. Tracking examples of the top four trackers. The symbol × means tracking failure. The bounding boxes are thinned for better visualization of the tracking effect differences, best viewed by zooming in.

5. Discussion

The experimental results demonstrate the effectiveness of the proposed approach in tracking rotating and non-rotating objects in SVs. Compared to those trackers that consider the rotation issue, the proposed method can perceive small angle deviations and provide more accurate orientation and size information. For example, the method of [37] can also detect the orientation of objects by computing the slope of object centroids. However, the slope defined in [37] may be difficult to represent the orientations of objects with obvious angle changes. The method of [40] uses the fixed-angle pool to solve object rotation and outputs the axis-aligned bounding boxes, which ignores accurate semantic information (e.g., real size, orientation of the object). Compared to [37,40], the proposed approach may be suitable to represent the orientations of objects with tiny and obvious angle changes and yield-rotating bounding boxes due to precise angle estimation effects of the rotation branch. In contrast to those trackers that handle the tracking drift issue, it guarantees precise localization assisted by the hybridization of angle and motion information. The methods of [38,39] also use optical flow for tracking. However, in [38,39], the Lucas–Kanade sparse optical flow [45] is regarded as a feature for representation, which may make it difficult to represent pixel-level motion information. In addition, [38,39] focus on considering the magnitudes of the optical flow and ignore the directional information. The proposed approach explores the effects of dense optical flow, which is capable of representing the pixel-level motion information. Furthermore, magnitudes coupled with directions of the optical flow are incorporated to prevent tracking drift and enhance the tracking performance.

The advantages of the proposed method are attributed to its branches (i.e., rotation and translation branches) in estimating the rotation angle and preventing tracking drift. In the estimation of rotation angle (i.e., rotation branch), this study attempts to reveal the relationship between the rotation and the translation. The rotation phenomenon is decomposed into the translation solution to achieve adaptive rotation estimation. In this way, the angle difference between adjacent frames can be obtained precisely by solving the translation problem. Figure 12 shows the effect of rotation angle estimation with a sample of the original (Figure 12a) and rotated patches (Figure 12b). In Figure 12b, the original ones are rotated to the initial orientation of the first frame by the estimated angle. It can ensure the spatial consistency of the adjacent patches as much as possible, allowing stable response results. Furthermore, the appearance and motion information contained in adjacent frames are synergized to enhance the object representations and deal with the tracking drift issue in the translation branch with motion constraint. To achieve the per-pixel motion constraint, the motion state of the object in the previous frame can be remapped to the current frame using the dense optical flow feature. Moreover, the proposed method can orient the bounding box to a more realistic object state with precise angle, size, and location information, and the results would serve a variety of scenarios (e.g., 2-D pose estimation of moving objects in a video, precise representations of dense objects in remote-sensing images, etc.).



(a)

(b)

Figure 12. Visual comparison of original patches (**a**) and rotated patches (**b**).

This paper verifies the significance of angle estimation and motion constraints for SOT in SVs. This work will help exploit the potential of satellite videos for applications such as traffic analysis, disaster response, and military target surveillance.

6. Conclusions

SOT in SVs has great potential for remote-sensing ground surveillance. To address object rotation and tracking drift problems, we analyzed the task from a new perspective, where the hybridization of angle and motion information cooperates for SOT in SVs. In addition, an RAMC tracker consisting of rotation and translation branches was created. By decomposing the rotation issue into translation solution, it decouples the rotation and translation motion patterns, achieving adaptive angle estimation. Subsequently, we dug out potential motion information and synergized it with the appearance information to prevent tracking drift. Moreover, an IS strategy was proposed to optimize the evaluation of trackers. Quantitative and qualitative experiments were conducted on space-born SV datasets. The results demonstrate that the proposed method yields state-of-the-art performance and runs at real-time speed. Future work will focus on solving the angle jitter problem.

Author Contributions: Conceptualization, Y.C. and Y.T.; methodology, Y.C., Y.T. and T.H.; resources, T.H. and Y.Z.; writing—original draft preparation, Y.C. and Y.T.; and supervision, Y.T., B.Z. and H.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (NSFC) (No. 41971313).

Acknowledgments: We are grateful to anonymous reviewers and Academic Editors for providing valuable comments. We thank Chang Guang Satellite Technology Co., Ltd. (Changchun, China) and the International Space Station for providing the satellite videos.

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

References

1. Makovski, T.; Vazquez, G.A.; Jiang, Y.V. Visual Learning in Multiple-Object Tracking. *PLoS ONE* **2008**, *3*, e2228. [CrossRef] [PubMed]
2. Xing, J.; Ai, H.; Lao, S. Multiple Human Tracking Based on Multi-view Upper-Body Detection and Discriminative Learning. In Proceedings of the 20th International Conference on Pattern Recognition, Istanbul, Turkey, 23–26 August 2010; pp. 1698–1701. [CrossRef]
3. Zhang, G.; Vela, P.A. Good Features to Track for VisuaL SLAM. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015. [CrossRef]
4. Smeulders, A.W.; Chu, D.M.; Cucchiara, R.; Calderara, S.; Dehghan, A.; Shah, M. Visual Tracking: An Experimental Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 1442–1468. [CrossRef] [PubMed]
5. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H.S. Fully-Convolutional Siamese Networks for Object Tracking. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2016; Volume 9914, pp. 850–865. [CrossRef]
6. Nam, H.; Han, B. Learning Multi-domain Convolutional Neural Networks for Visual Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4293–4302. [CrossRef]
7. Li, B.; Yan, J.; Wu, W.; Zhu, Z.; Hu, X. High Performance Visual Tracking with Siamese Region Proposal Network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8971–8980. [CrossRef]
8. 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 IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 4277–4286. [CrossRef]
9. Chen, Z.; Zhong, B.; Li, G.; Zhang, S.; Ji, R. Siamese Box Adaptive Network for Visual Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 6667–6676. [CrossRef]
10. 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] [PubMed]
11. Bertinetto, L.; Valmadre, J.; Golodetz, S.; Miksik, O.; Torr, P.H.S. Staple: Complementary Learners for Real-Time Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1401–1409. [CrossRef]
12. Galoogahi, H.K.; Fagg, A.; Lucey, S. Learning Background-Aware Correlation Filters for Visual Tracking. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1144–1152. [CrossRef]
13. Lukezic, A.; Vojir, T.; Zajc, L.C.; Matas, J.; Kristan, M. Discriminative Correlation Filter Tracker with Channel and Spatial Reliability. *Int. J. Comput. Vis.* **2018**, *126*, 671–688. [CrossRef]
14. Hong, S.; You, T.; Kwak, S.; Han, B. Online Tracking by Learning Discriminative Saliency Map with Convolutional Neural Network. In Proceedings of the International Conference on Machine Learning, Lille, France, 7–9 July 2015; pp. 597–606. [CrossRef]
15. Cortes, C.; Vapnik, V. Support Vector Networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
16. Nam, H.; Baek, M.; Han, B. Modeling and Propagating CNNs in a Tree Structure for Visual Tracking. *arXiv* **2016**, arXiv:1608.07242. [CrossRef]
17. Wang, Q.; Zhang, L.; Bertinetto, L.; Hu, W.; Torr, P.H.S. Fast Online Object Tracking and Segmentation: A Unifying Approach. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 1328–1338. [CrossRef]
18. Danelljan, M.; Hager, G.; Khan, F.S.; Felsberg, M. Convolutional Features for Correlation Filter Based Visual Tracking. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Santiago, Chile, 11–18 December 2015; pp. 621–629. [CrossRef]

19. 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, Amsterdam, The Netherlands, 8–16 October 2016; pp. 472–488. [CrossRef]
20. Danelljan, M.; Bhat, G.; Khan, F.S.; Felsberg, M. ECO: Efficient Convolution Operators for Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6931–6939. [CrossRef]
21. Bolme, D.S.; Beveridge, J.R.; Draper, B.A.; Lui, Y.M. Visual Object Tracking using Adaptive Correlation Filters. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 2544–2550. [CrossRef]
22. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. Exploiting the Circulant Structure of Tracking-by-Detection with Kernels. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7575, pp. 702–715. [CrossRef]
23. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–25 June 2005; pp. 886–893. [CrossRef]
24. Li, Y.; Liu, G. Learning a Scale-and-Rotation Correlation Filter for Robust Visual Tracking. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 454–458. [CrossRef]
25. Rout, L.; Siddhartha; Mishra, D.; Gorthi, R. Rotation Adaptive Visual Object Tracking with Motion Consistency. In Proceedings of the 18th IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahou, NV/CA, USA, 12–15 March 2018; pp. 1047–1055. [CrossRef]
26. Li, Y.; Zhu, J.; Hoi, S.C.H.; Song, W.; Wang, Z.; Liu, H.; Aaai. Robust Estimation of Similarity Transformation for Visual Object Tracking. In Proceedings of the 33rd AAAI Conference on Artificial Intelligence/31st Innovative Applications of Artificial Intelligence Conference/9th AAAI Symposium on Educational Advances in Artificial Intelligence, Honolulu, HI, USA, 27 February–1 March 2019; pp. 8666–8673. [CrossRef]
27. Danelljan, M.; Hager, G.; Khan, F.S.; Felsberg, M. Learning Spatially Regularized Correlation Filters for Visual Tracking. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 11–18 December 2015; pp. 4310–4318. [CrossRef]
28. Zhang, G. Satellite video processing and applications. *J. Appl. Sci.* **2016**, *34*, 361–370. [CrossRef]
29. Wang, Y.M.; Wang, T.Y.; Zhang, G.; Cheng, Q.; Wu, J.Q. Small Target Tracking in Satellite Videos Using Background Compensation. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 7010–7021. [CrossRef]
30. 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]
31. Zhu, K.; Zhang, X.D.; Chen, G.Z.; Tan, X.L.; Liao, P.Y.; Wu, H.Y.; Cui, X.J.; Zuo, Y.A.; Lv, Z.Y. Single Object Tracking in Satellite Videos: Deep Siamese Network Incorporating an Interframe Difference Centroid Inertia Motion Model. *Remote Sens.* **2021**, *13*, 1298. [CrossRef]
32. Yin, Z.Y.; Tang, Y.Q. Analysis of Traffic Flow in Urban Area for Satellite Video. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Waikoloa, HI, USA, 26 September–2 October 2020; pp. 2898–2901. [CrossRef]
33. Li, W.T.; Gao, F.; Zhang, P.; Li, Y.H.; An, Y.; Zhong, X.; Lu, Q. Research on Multiview Stereo Mapping Based on Satellite Video Images. *IEEE Access* **2021**, *9*, 44069–44083. [CrossRef]
34. Legleiter, C.J.; Kinzel, P.J. Surface Flow Velocities From Space: Particle Image Velocimetry of Satellite Video of a Large, Sediment-Laden River. *Front. Water* **2021**, *3*, 652213. [CrossRef]
35. Ao, W.; Fu, Y.; Hou, X.; Xu, F. Needles in a Haystack: Tracking City-Scale Moving Vehicles from Continuously Moving Satellite. *IEEE Trans. Image Process.* **2019**, *29*, 1944–1957. [CrossRef] [PubMed]
36. Xuan, S.Y.; Li, S.Y.; Han, M.F.; 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]
37. Guo, Y.J.; Yang, D.Q.; Chen, Z.Z. Object Tracking on Satellite Videos: A Correlation Filter-Based Tracking Method with Trajectory Correction by Kalman Filter. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 3538–3551. [CrossRef]
38. Shao, J.; Du, B.; Wu, C.; Zhang, L.F. Tracking Objects from Satellite Videos: A Velocity Feature Based Correlation Filter. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 7860–7871. [CrossRef]
39. Shao, J.; Du, B.; Wu, C.; Zhang, L. Can We Track Targets From Space? A Hybrid Kernel Correlation Filter Tracker for Satellite Video. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 8719–8731. [CrossRef]
40. Xuan, S.Y.; Li, S.Y.; Zhao, Z.F.; Zhou, Z.; Zhang, W.F.; Tan, H.; Xia, G.S.; Gu, Y.F. Rotation Adaptive Correlation Filter for Moving Object Tracking in Satellite Videos. *Neurocomputing* **2021**, *438*, 94–106. [CrossRef]
41. Liu, Y.S.; Liao, Y.R.; Lin, C.B.; Jia, Y.T.; Li, Z.M.; Yang, X.Y. Object Tracking in Satellite Videos Based on Correlation Filter with Multi-Feature Fusion and Motion Trajectory Compensation. *Remote Sens.* **2022**, *14*, 777. [CrossRef]
42. Chen, Y.Z.; Tang, Y.Q.; Yin, Z.Y.; Han, T.; Zou, B.; Feng, H.H. Single Object Tracking in Satellite Videos: A Correlation Filter-Based Dual-Flow Tracker. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, 1–13. [CrossRef]
43. Kalman, R.E. A New Approach to Linear Filtering and Prediction Problems. *J. Basic Eng.* **1960**, *82*, 35–45. [CrossRef]
44. 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]
45. Patel, D.; Upadhyay, S. Optical Flow Measurement Using Lucas Kanade Method. *Int. J. Comput. Appl.* **2013**, *61*, 6–10. [CrossRef]

46. Xu, L.; Jia, J.; Matsushita, Y. Motion Detail Preserving Optical Flow Estimation. *IEEE Trans. Pattern Anal. Mach. Intell* **2012**, *34*, 1744–1757. [CrossRef]
47. Reddy, B.S.; Chatterji, B.N. An FFT-Based Technique for Translation, Rotation, and Scale-Invariant Image Registration. *IEEE Trans. Image Process.* **1996**, *5*, 1266–1271. [CrossRef]
48. Nagel, H.H.; Enkelmann, W. An Investigation of Smoothness Constraints for The Estimation of Displacement Vector Fields from Image Sequences. *IEEE Trans. Pattern Anal. Mach. Intell* **1986**, *8*, 565–593. [CrossRef]
49. Farnebäck, G. Two-Frame Motion Estimation Based on Polynomial Expansion. In Proceedings of the Scandinavian Conference on Image Analysis, Halmstad, Sweden, 29 June–2 July 2003; pp. 363–370. [CrossRef]
50. Wu, Y.; Lim, J.; Yang, M.-H. Online Object Tracking: A Benchmark. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 2411–2418.
51. Wu, Y.; Lim, J.; Yang, M.H. Object Tracking Benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1834–1848. [CrossRef]
52. Du, B.; Cai, S.H.; Wu, C. Object Tracking in Satellite Videos Based on a Multiframe Optical Flow Tracker. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 3043–3055. [CrossRef]
53. Li, Y.; Zhu, J. A Scale Adaptive Kernel Correlation Filter Tracker with Feature Integration. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2015; pp. 254–265. [CrossRef]
54. Li, F.; Tian, C.; Zuo, W.; Zhang, L.; Yang, M.H. Learning Spatial-Temporal Regularized Correlation Filters for Visual Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018. [CrossRef]
55. Dai, K.; Wang, D.; Lu, H.; Sun, C.; Li, J. Visual Tracking via Adaptive Spatially-Regularized Correlation Filters. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019. [CrossRef]
56. Xu, T.Y.; Feng, Z.H.; Wu, X.J.; Kittler, J. Joint Group Feature Selection and Discriminative Filter Learning for Robust Visual Object Tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 7949–7959. [CrossRef]
57. Li, Y.; Fu, C.; Ding, F.; Huang, Z.; Lu, G. AutoTrack: Towards High-Performance Visual Tracking for UAV with Automatic Spatio-Temporal Regularization. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020. [CrossRef]
58. Ren, S.Q.; He, K.M.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Proceedings of the 29th Annual Conference on Neural Information Processing Systems (NIPS), Montreal, Canada, 7–12 December 2015. [CrossRef]