Satellite video single object tracking: A systematic review and an oriented object tracking benchmark

Yuzeng Chen, Yuqi Tang, Yi Xiao, Qiangqiang Yuan, Yuwei Zhang, Fengqing Liu, Jiang He, Liangpei Zhang

Abstract

Single object tracking (SOT) in satellite video (SV) enables the continuous acquisition of position and range information of an arbitrary object, showing promising value in remote sensing applications. However, existing trackers and datasets rarely focus on the SOT of oriented objects in SV. To bridge this gap, this article presents a comprehensive review of various tracking paradigms and frameworks covering both the general video and satellite video domains and subsequently proposes the oriented object tracking benchmark (OOTB) to advance the field of visual tracking. OOTB contains 29,890 frames from 110 video sequences, covering common satellite video object categories including car, ship, plane, and train. All frames are manually annotated with oriented bounding boxes, and each sequence is labeled with 12 fine-grained attributes. Additionally, a high-precision evaluation protocol is proposed for comprehensive and fair comparisons of trackers. To validate the existing trackers and explore frameworks suitable for SV tracking, we benchmark 33 state-of-the-art trackers totaling 58 models with different features, backbones, and tracker tags. Finally, extensive experiments and insightful thoughts are also provided to help understand their performance and offer baseline results for future research. OOTB is available at https://github.com/YZCU/OOTB.

1. Introduction

Single object tracking (SOT) is one of the most essential tasks in computer vision, which allows the establishment of object correspondences in video sequences (Javed et al., 2022). Given the initial state, SOT aims to determine subsequent states of an arbitrary object. SOT can be applied to a variety of fields such as autonomous driving, intelligent surveillance, robotics, and augmented reality. Tracking technology has received a lot of attention, and many advanced trackers have been proposed to solve realistic challenges such as scale variation, deformation, similar appearance, and illumination changes (Chen et al., 2023b). With the advancement of trackers, the tracking benchmark plays a fundamental role in performance evaluation (Wang et al., 2022a). Several widely used benchmarks such as LaSOT (Fan et al., 2019), TrackingNet (Muller et al., 2018), and LasHeR (Li et al., 2022a) have been released for evaluating trackers and promoting the development of visual tracking.

Satellite video (SV) is a valuable surface observation data that provides a wealth of static and dynamic information on specific areas (Feng et al., 2021). In 2013, the SkySat-1 (SS) satellite captured a panchromatic video with a ground sample distance (GSD) of 1.1 m and a frame rate of 30 frames per second (FPS). In 2016, the International Space Station (ISS) captured a 3 FPS red–green–blue (RGB) video with a GSD of 1.0 m. The Jilin-1 (JL) satellite constellation can capture 30 FPS RGB video sequences with a 0.92 m GSD. Recently, the Luojia-3–01 satellite was launched, which has the capability of multi-mode optical imaging, intelligent processing in orbit, and real-time transmission in star-to-earth and star-to-star communication. Table 1 presents the detailed configurations and parameters of some video satellites, and Fig. 1 shows the sample frames corresponding to these satellites. The emergence of SV data enhances remote sensing observation capabilities and facilitates the visual tracking community (Wu et al., 2022). SOT in SV has promising applications in intelligent traffic surveillance and analysis (Du et al., 2018), etc. As mentioned above, remarkable advances have been...
made in the SOT of generic video (GV). GV can be captured by commonly used devices such as closed-circuit televisions and unmanned aerial vehicles (UAVs) (Wang et al., 2020). In contrast, progress in SV object tracking still lags far behind that of GV due to the lack of well-annotated benchmark datasets and evaluation protocols. It is also difficult to achieve accurate and robust tracking due to the following challenges:

- When it comes to SOT in SV, existing high-quality and public datasets and benchmarks are insufficient. There are rarely available datasets with oriented bounding box (OBB) annotations for single object tracking, which are essential for accurately tracking oriented objects. Additionally, it is fundamental to measure the performance of trackers comprehensively and fairly, particularly for OBB annotations with various sizes and uneven aspect ratios.
- SV typically contains three bands (i.e., red, green, and blue), which results in limited spectral features of objects, as shown in Fig. 1. Furthermore, moving objects are often small and occupy a few pixels resulting in limited spatial features such as context and texture. This can make it difficult to accurately estimate the object state, as demonstrated in Fig. 2.
- SV is photographed by the high-speed moving satellite platform. Accompanied by non-stationary and complex backgrounds, small objects are susceptible to abnormal interferences such as similar appearance, partial occlusion, motion blur, and background clutters, as shown in Fig. 3.

This article establishes the first available oriented object tracking benchmark (OOTB) for SOT in SV. OOTB includes 110 sequences with a total of 29,890 frames, covering common object categories. Moreover, a high-precision evaluation protocol is proposed to achieve comprehensive and fair comparisons of trackers. We also benchmark 33 state-of-the-art (SOTA) trackers with a total of 58 models covering different features, backbones, and tracker tags to help understand their performance and offer baseline results for future research. Extensive comparisons and analysis demonstrate that SV object tracking remains challenging in the visual tracking community. The major contributions are summarized as follows:

- We provide a comprehensive and detailed review of various tracking paradigms and frameworks, covering both the general video and satellite video domains. We also present relevant single object tracking benchmarks for generic and specific applications.
- We construct an oriented object tracking benchmark OOTB. To the best of our knowledge, OOTB is the first available oriented benchmark dedicated to SOT in SV. It consists of 110 sequences totaling 29,890 frames and covers common object categories including car, ship, plane, and train. All sequences are manually annotated with high-quality oriented bounding boxes and labeled with 12 fine-grained attributes, making them an invaluable resource for relevant research. In addition, we propose a high-precision evaluation protocol for fair comparisons between trackers.
- We benchmark 33 SOTA trackers with a total of 58 models, covering various tracking paradigms and application scenarios. Moreover, the in-depth comparison and analysis are conducted to provide baseline results for further research. In light of the advances in trackers, several insightful thoughts are also drawn to point out promising prospects in the SV tracking domain.

The rest of this article is organized as follows. Sections 2-4 give a comprehensive review including GV trackers, SV trackers, and benchmark datasets. The proposed OOTB and compared trackers are introduced in Section 5. Experimental results and analysis are presented in Section 6. In Section 7, we provide several thoughts and insights for future research. Finally, we conclude this article and summarize the contributions.

2. Review on generic video trackers

Depending on the data acquisition platform, SOT can be divided into GV and SV tracking domains. In this article, we provide a comprehensive overview of trackers and techniques including advancements, challenges, and limitations in GV and SV object tracking. We also outline relevant benchmarks that are commonly used to evaluate the performance of SOT trackers.

Typically, SOT trackers can be classified into two categories: generative paradigm and discriminative paradigm (You et al., 2019). The former constructs a model to represent the object and finds an object region that is similar to the description of the generative model by minimizing the signal and minimizing the objective loss. Object representation models such as Gaussian mixed model (Jepson et al., 2003), kernel trick (Han et al., 2008), and sparse representation (Wright et al., 2010) can directly affect the accuracy and speed of tracking methods in the generative paradigm. On the other hand, the discriminative paradigm jointly trains foreground and background regions to discriminate the object, which improves the tracking robustness. Its simplicity and strong performance have made it a fundamental paradigm for tracking in recent decades (You et al., 2019). Discriminative correlation filter (DCF) and Siamese neural network (SNN), two of the best-performing discriminative paradigms, have proven their advancement and are dominating the SOT domain (Javed et al., 2022).

Moreover, other paradigms, such as Transformer, recurrent neural network (RNN), generative adversarial network (GAN), and traditional convolutional neural network (CNN), have also achieved satisfactory results in the tracking community (Marvasti-Zadeh et al., 2022). Next, we provide a comprehensive overview of these paradigms.

2.1. DCF for SOT

Over the last decade, DCFs have proved their high performance and efficiency on various benchmarks (Javed et al., 2022). The DCF learns a filter by minimizing a least-squares error to determine the object’s position and updates the model to adapt to object changes during tracking. In the following, we review the DCF in terms of discriminative object representations, adaptive scale estimation, and handling boundary effects.
2.1.1. Discriminative object representations

DCF-based trackers have been a highlight since the introduction of MOSSE (Bolme et al., 2010). Subsequently, CSK (Henriques et al., 2012) modeled after MOSSE introduces the circular matrix and kernel trick to improve tracking performance. Both MOSSE and CSK use intensity or raw-pixel features to represent objects.

Considering the limitations of intensity or raw pixel features, several color-based features, e.g., local color, color histogram (CH), and color name have also been explored for enhanced object representations. Representative DCF-based trackers are LCT (Ma et al., 2015b), CN (Danelljan et al., 2014), DAT (Possegger et al., 2015), and Staple (Bertinetto et al., 2016a). The CN tracker extends the training mechanism of CSK to multi-channel color name features and proposes an adaptive dimensionality reduction method, which proves the significance of seriously selecting color transformations. Color name features have been employed in lots of DCF-based trackers such as CSR-DCF (Lukezic et al., 2018), ECO-HC (Danelljan et al., 2017a), ARCF (Huang et al., 2019), GFS-DCF (Xu et al., 2019), and AutoTrack (Li et al., 2020). Another powerful hand-crafted feature is the histogram of oriented gradients (HOG) that is initially designed for pedestrian detection (Dalal and Triggs, 2005). Due to its advantages in capturing contours and remaining intrinsic illumination invariance, KCF (Henriques et al., 2015) extends CSK with multi-channel HOG features and introduces multiple kernel functions to train more discriminative classifiers.

![Sample frames from video satellites. (a), (b), (c), (d), and (e) are captured by the SkySat-1, ISS, Luojia-3-01, OVS-1, and Jilin-1, respectively.](image)

Fig. 1. Sample frames from video satellites. (a), (b), (c), (d), and (e) are captured by the SkySat-1, ISS, Luojia-3-01, OVS-1, and Jilin-1, respectively.
Fig. 2. Visual examples of SV objects. (a) shows the original frame, while (b) and (c) are two zoomed-in regions. (b) and (c) display the car and plane categories, marked by yellow boxes. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
number of DCF-based trackers utilize deep convolutional neural nets (Li et al., 2019c).

Multiple feature fusion strategy has been validated and adopted by illumination variation, and deformation (Bertinetto et al., 2016a).

emerging. For instance, Staple combines HOG and CH with a ridge spatial layout of the object (Dalal and Triggs, 2005). Therefore, combining complementary features to cope with multiple challenges is emerging. For instance, Staple combines HOG and CH with a ridge regression framework, achieving robust tracking under color change, illumination variation, and deformation (Bertinetto et al., 2016a).

Multiple feature fusion strategy has been validated and adopted by several DCF-based trackers such as SAMF (Li and Zhu, 2015), Staple (Bertinetto et al., 2016a), STRCF (Li et al., 2018b), MCCT (Wang et al., 2018), AutoTrack (Li et al., 2020), GFS-DCF (Xu et al., 2019), and LDES (Li et al., 2019c).

Encouraged by recent advances in deep learning, an increasing number of DCF-based trackers utilize deep convolutional neural networks (CNNs) that are suitable for image processing tasks (Ye et al., 2022). Shallow CNN features comprise low-level information with high spatial resolution, suitable for accurate object localization, while deep features encode high-level semantic information with low resolution, inherently invariant to appropriate object changes. HCF (Ma et al., 2015a) tracker, as one of the earliest DCF-based trackers to use CNN features, explores features of different dimensions to represent objects and trains a multi-resolution filter to locate the object in a coarse-to-fine fashion. Other DCF-based trackers such as CFNet (Valmadre et al., 2017), DeepSRDCF (Danelljan et al., 2015a), ECO (Danelljan et al., 2017a), ASRFC (Oai et al., 2019), ARCF (Huang et al., 2019), ATOM (Danelljan et al., 2019b), DIMP (Bhat et al., 2015), and PrDImp (Danelljan et al., 2020) have also demonstrated the effectiveness of CNN features on various benchmarks, paving the way for exploring more sophisticated trackers.

2.1.2. Adaptive scale estimation

Blending features greatly contribute to the accuracy and robustness of DCF-based trackers. However, the tracked object usually suffers from position and scale changes. Standard DCF-based trackers use a fixed-size template and are unable to handle scale changes, leading to severe tracking drifts (Javed et al., 2021). Towards this end, several strategies have been investigated for accurate scale estimation. SAMF (Li and Zhu, 2015) defines a scaling pool that acquires multi-resolution scaling patches to estimate the position and scale of objects. Nevertheless, this method is computationally expensive. Considering small and moderate scale variations in neighboring frames, DSST (Danelljan et al., 2017) first estimates the object position using a two-dimensional filter and then uses a one-dimensional filter for scale estimation. Due to its efficiency and effectiveness, this strategy has been utilized in various trackers, e.g., BACF (Galoogahi et al., 2017b), LCT (Ma et al., 2015b), Staple (Bertinetto et al., 2016a), CACF (Mueller et al., 2017), CSR-DCF (Lukezic et al., 2018), and MCCT (Wang et al., 2018). In recent SOTA trackers, the deep bounding box regression approach has shown appealing results without manually setting the scale estimation parameters. It has become a universal component in DCF-based trackers such as DImp (Bhat et al., 2019), ATOM (Danelljan et al., 2019b), PrDiMP (Danelljan et al., 2020), KYS (Bhat et al., 2020), and KeepTrack (Mayer et al., 2021).

2.1.3. Handling boundary effects

In the evolution of DCF-based trackers, the boundary effect caused by the periodic assumption of training samples is a stubborn stumbling block that severely limits the search region and degrades the discrimination capability of models (Wang et al., 2022a). Several solutions have been proposed to overcome this issue in numerous DCF-based trackers such as CFLB (Galoogahi et al., 2015), SRDCF (Danelljan et al., 2015), BACF (Galoogahi et al., 2017b), CSR-DCF (Lukezic et al., 2018), STRCF (Li et al., 2018b), ARCF (Huang et al., 2019), ASRFC (Oai et al., 2019), AutoTrack (Li et al., 2020), GFS-DCF (Xu et al., 2019), DRCF (Fu et al., 2020), ATOM (Danelljan et al., 2019b), and DIMP (Bhat et al., 2019). For example, CFLB (Galoogahi et al., 2015) trains filters with few samples to attenuate boundary effects. SRDCF (Danelljan et al., 2015) introduces a spatial regularization function that penalizes filter coefficients so that filters can be trained for large regions. For efficiency, BACF successfully trains a background-aware filter from real negative samples densely sampled from backgrounds. CSR-DCF (Lukezic et al., 2018) applies spatial-domain constraints to the filter to weaken the influence of boundary effects. The aforementioned trackers have made great progress in addressing boundary effects and advancing DCF development. With continuous improvements, SOTA DCF-based trackers such as ATOM (Danelljan et al., 2019b) and DIMP (Bhat et al., 2019) can circumvent the issue by directly learning a filter in the spatial domain.

2.2. SNN for SOT

Conventionally, SNN-based trackers consist of two branches: the template branch and the candidate branch. The template branch takes as input the image patch of the first frame or previous frames, while the candidate branch receives the image patches of the subsequent frames. Both branches share a CNN trained from massive sample pairs to ensure that the same transformation is imposed on these two branches (Javed et al., 2022). Due to its superior performance and efficiency, SNN has aroused extensive attention in the visual tracking community. Given a large number of training sample pairs, the SNN-based tracker is capable of...
of learning the general relationship between object appearance and motion and also locating unseen objects in the training set. The primary objective of SNN-based trackers is to overcome the limitations of pre-trained deep neural networks and fully leverage end-to-end training for real-time applications (Marvasti-Zadeh et al., 2022). In this section, we review the evolution of SNN-based trackers from discriminative object representation, adaptive scale estimation, and balancing training data.

2.2.1. Discriminative object representation

Robust object representations are fundamental for reliable tracking, and discriminative object models rely heavily on the backbone network. One of the pioneers of SNN-based trackers, SiamFC (Bertinetto et al., 2016), has fine-tuned the pre-trained AlexNet (Krizhevsky et al., 2017) parameters for visual tracking, and experimental results have shown its superiority over the DCF-based trackers at that time. Many SNN-based trackers, e.g., GOTURN (Ilie et al., 2016), SINT (Tao et al., 2016), SiamRPN (Li et al., 2018a), DaSiamRPN (Zhu et al., 2018), C-RPN (Fan et al., 2019b), and SA-Siam (He et al., 2018), also integrate AlexNet as a feature extractor. To achieve better results, SiamRPN introduces the regional proposal network (RPN) (Ren et al., 2017) for proposal generation. However, the AlexNet structure is relatively shallow, making it difficult to extract stronger and more powerful features when compared to deeper networks such as VGGNet (Chatfield et al., 2014; Simonyan and Zisserman, 2014), GoogLeNet (Szegedy et al., 2015), ResNet (He et al., 2016), and ResNetXt (Xie et al., 2017). Therefore, exploring how to exploit a deeper and wider network as a backbone is crucial for enhancing discrimination. SiamRPN++ (Li et al., 2019a) driven by ResNet breaks the restriction of translational invariance, enabling more accurate and robust tracking. SiamDW (Zhang, 2019b) uses deeper and wider backbones including VGGNet, ResNet, and Inception. Both SiamRPN++ and SiamDW have proven their superiority on various benchmarks. Building on previous work, SOTA trackers such as SiamMask (Wang et al., 2019c), D3S (Lakezic et al., 2020), SiamGAT (Guo et al., 2021), TransIT (Chen et al., 2021), KeepTrack (Mayer et al., 2021), Stark (Yan et al., 2021a), SiamCAR (Cui et al., 2022a), and AiATrack (Gao et al., 2022) are continuously exploring the potential of deeper and wider backbones. However, these SOTA trackers are always dependent on hand-crafted models. To address the issue, LightTrack (Yan et al., 2021) embeds automatically designed lightweight models using the neural architecture search (NAS) (Pham et al., 2018) therefore performing effectively and efficiently. Such customized models can bridge the gap between academia and industry and are expected to identify forward-looking directions in the coming years.

2.2.2. Adaptive scale estimation

Similar to DCF-based trackers, SNN-based trackers also encounter object scale variations. In the early years, SNN-based trackers use common multiple-resolution scale search approaches to handle scale variations. For example, SiamFC searches for multiple scales in the forward pass by integrating a mini-batch of scaled patches. Due to its simplicity, this approach has been utilized in several SNN-based trackers such as SA-Siam (He et al., 2018), TADT (Li et al., 2019b), UDT (Wang et al., 2019b), and FlowTrack (Zhu et al., 2018b). However, the multiple-resolution scale search approach suffers from expensive computational costs. Inspired by scale estimation in object detection, Siamese trackers introduce RPN to predict region proposals with relative scales and aspect ratios (Ren et al., 2017). The RPN consists of a classification network for estimating foreground-background and a regression network for refining anchor boxes. SiamRPN attaches an RPN to the Siamese network to extract region proposals, discarding the time-consuming multi-scale search. Experimental results have shown the superiority of RPN in accuracy and efficiency. RPN has become a fundamental anchor-based bounding box regression approach among various SNN-based trackers, e.g., SiamRPN++ (Li et al., 2019a), DaSiamRPN, SiamMask (Wang et al., 2019c), SiamDW, SPM (Wang et al., 2019a), C-RPN (Fan et al., 2019b), and SwinTrack (Lin et al., 2022). The anchor-free bounding box regression approach is popular in the detection community due to its simplified structure and lack of dependence on hyperparameters (Cui et al., 2022a). Typically, there are two types of bounding box solutions for anchor-free methods, i.e., center-based (Tian et al., 2019) and keypoint-based (Law and Deng, 2020) algorithms. The former directly estimates the object’s center and the distance from the center to the boundary. The latter (i.e., the keypoint-based approach) detects the top-left and bottom-right corner positions to form a bounding box. Motivated by the center-based detection strategy, Ocean (Zhang et al., 2020) implements the prediction of the object position and scale in an anchor-free fashion. SNN-based trackers such as SiamBAN (Chen et al., 2020), SiamCAR (Cui et al., 2022a), SiamFC++ (Xu et al., 2020), Stark (Yan et al., 2021a), ODTrack (Zheng et al., 2024), and MixFormer (Cui et al., 2024) have also inherited anchor-free bounding box regression, which is expected to be a popular alternative.

2.2.3. Balancing training data

Training data is crucial for improving model robustness. Some large-scale datasets such as ALOV300++ (Smellders et al., 2014), MSCOCO (Lin et al., 2014), ILSVRC-DET (Russakovsky et al., 2015), ILSVRC-VID (Russakovsky et al., 2015), NUS-PRO (Li et al., 2016), UAV123 (Mueller et al., 2016), YouTube-VOS (Xu et al., 2018), TrackingNet (Muller et al., 2018), GOT-10k (Huang et al., 2021), and LaSOT (Fan et al., 2019) have been used to train SNN-based trackers offline. However, there are far fewer positive samples than negative samples in offline training. The imbalanced distribution of training samples may seriously affect the discriminative ability of the model. To this end, several strategies have been investigated in Siamese trackers. DaSiamRPN integrates hard negative sampling to introduce more semantic negative sample pairs from the same and different categories. This strategy allows DaSiamRPN to focus on fine-grained object representations, attenuating tracking drifts. C-RPN (Fan et al., 2019b) cascades a sequence of RPNs to stimulate hard negative sampling and progressively refine bounding boxes. Other trackers, such as ATOM (Danelian et al., 2019b), TADT (Li et al., 2019b), and UDT (Wang et al., 2019b), inherited the correlation filter in the Siamese structure, strive to balance the training data and achieve competitive performance.

2.3. Transformer for SOT

Transformer (Vaswani et al., 2017) is an architecture that transforms one sequence into another using attention-based encoders and decoders. Recently, Transformer trackers have made remarkable progress, which can be classified into two categories including CNN-Transformer trackers and Fully-Transformer trackers (Kugarajeevan et al., 2023).

The former inherits the SNN paradigm and partially uses the Transformer architecture. More concretely, the CNN-Transformer tracker utilizes CNN, such as AlexNet (Krizhevsky et al., 2017), ResNet (He et al., 2016), and ShuffleNetV2 (Ma et al., 2018), to extract deep features of the template and candidate regions, followed by using the Transformer mechanism to achieve feature interactions. Finally, the prediction head receives features generated by the Transformer for localization. Benefiting from the Transformer architecture, CNN-transformer trackers can capture the non-linear interactions between the template and candidate regions, resulting in superior tracking performance, as demonstrated in TransIT (Chen et al., 2021), Stark (Yan et al., 2021a), TrDiMP (Wang et al., 2021), AIATrack (Gao et al., 2022), SiamTPN (Xing et al., 2022), ToMP (Mayer et al., 2022), etc.

However, they still rely on CNN for feature extraction, which uses the local convolution kernel to capture features. Therefore, it is difficult for CNN-Transformer trackers to capture global feature representations. To address this issue, the latter (i.e., the Fully-Transformer tracker) has been developed, which can be further categorized into the two-stream two-stage paradigm and one-stream one-stage paradigm. The two-
Table 2: Characteristics and experiments of some SV trackers. These trackers are listed in chronological order.

<table>
<thead>
<tr>
<th>Tracker</th>
<th>Exploited features @ Tracker prototype</th>
<th>RTFO</th>
<th>PC (CPU, RAM, Nvidia GPU)</th>
<th>Data source</th>
<th>Tracked object</th>
<th>Tracking performance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Category/NoSO</td>
<td>Benchmark/AO/AUC_S/AUC_P/EAO/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CPU/GPU</td>
</tr>
<tr>
<td>KCF, TFD (Du et al., 2018)</td>
<td>HOG + MFD @ KCF</td>
<td>—</td>
<td>Intel I5 2.8 GHz CPU, 8 GB RAM</td>
<td>JL, ISS</td>
<td>C, T</td>
<td>3 HBB OTB</td>
</tr>
<tr>
<td>MOFT (Du et al., 2019)</td>
<td>OF @ I</td>
<td></td>
<td></td>
<td>JL, ISS</td>
<td>C, P, T</td>
<td>5 HBB OTB</td>
</tr>
<tr>
<td>CF (Guo et al., 2019)</td>
<td>HOG + I + PM @ DSST</td>
<td>APCE + MFD</td>
<td>Intel I7-3770 3.4 GHz CPU, 32 GB RAM</td>
<td>SS, JL</td>
<td>C</td>
<td>31 OBB VOT</td>
</tr>
<tr>
<td>MOFT (Shao et al., 2019a)</td>
<td>DA + BS + PM @ SiamFC</td>
<td>GMM + MFD</td>
<td></td>
<td>JL, ISS</td>
<td>C, T</td>
<td>3 HBB OTB</td>
</tr>
<tr>
<td>MOFT (Shao et al., 2019b)</td>
<td>KCF</td>
<td>OF @ KCF</td>
<td></td>
<td>JL, ISS</td>
<td>C, P, T</td>
<td>6 HBB OTB</td>
</tr>
<tr>
<td>VCF (Shao et al., 2019c)</td>
<td>—</td>
<td></td>
<td></td>
<td>JL, ISS</td>
<td>P, T</td>
<td>3 HBB OTB</td>
</tr>
<tr>
<td>CRAM (Hu et al., 2020)</td>
<td>DA + DM @ (Danelljan et al., 2019)</td>
<td>—</td>
<td>3.5 GHz CPU, GTX 1080 GPU</td>
<td>JL, SS</td>
<td>C</td>
<td>31 HBB VOT</td>
</tr>
<tr>
<td>WTIC (Wang et al., 2020)</td>
<td>G + PM @ CSK</td>
<td></td>
<td></td>
<td>JL</td>
<td>C</td>
<td>9 HBB OTB</td>
</tr>
<tr>
<td>CF (Xuan et al., 2020)</td>
<td>HOG + PM @ KCF</td>
<td>APCE + MFD</td>
<td>Intel Xeon E5-2620v3 2.4 GHz CPU</td>
<td>JL</td>
<td>C, P</td>
<td>13 HBB OTB</td>
</tr>
<tr>
<td>VAAH (Li et al., 2021)</td>
<td>DA @ SiamFC</td>
<td></td>
<td></td>
<td>JL, SS</td>
<td>C, S, P</td>
<td>80 OBB</td>
</tr>
<tr>
<td>CF, FFMC (Liu et al., 2021)</td>
<td>HOG + LBP + I + PM @ KCF</td>
<td>APCE + MFD</td>
<td>Intel Xeon E3-1240v5 3.50 GHz CPU</td>
<td>JL</td>
<td>C, S, P</td>
<td>&gt;10 HBB OTB</td>
</tr>
<tr>
<td>HR-Siam (Shao et al., 2021)</td>
<td>DA @ SiamRPN</td>
<td></td>
<td></td>
<td>JL, ISS</td>
<td>C, P, T</td>
<td>6 HBB OTB</td>
</tr>
<tr>
<td>RACF (Xuan et al., 2021)</td>
<td>HOG @ KCF</td>
<td></td>
<td></td>
<td>JL</td>
<td>C</td>
<td>6 HBB OTB</td>
</tr>
<tr>
<td>ID-DSN (Zhu et al., 2021)</td>
<td>DA + PM @ SiamRPN</td>
<td></td>
<td></td>
<td>JL, ISS</td>
<td>C, S, P, T</td>
<td>6 HBB OTB</td>
</tr>
<tr>
<td>HMTS (Chen et al., 2022b)</td>
<td>I + CN + PM @ KCF</td>
<td></td>
<td></td>
<td>JL, SS*</td>
<td>C, 65</td>
<td>HBB OTB</td>
</tr>
<tr>
<td>DF (Chen et al., 2022)</td>
<td>HOG + CN + GCS + PM @ Staple</td>
<td></td>
<td></td>
<td>JL</td>
<td>C</td>
<td>14 HBB OTB</td>
</tr>
<tr>
<td>RAMC (Chen et al., 2022b)</td>
<td>HOG + OF @ KCF</td>
<td></td>
<td></td>
<td>JL, ISS</td>
<td>C, S, P, T</td>
<td>8 OBB OTB</td>
</tr>
<tr>
<td>SRF, JMM (Li and Bian, 2022)</td>
<td>HOG + CN + PM @ STC</td>
<td></td>
<td></td>
<td>JL, SS*</td>
<td>C, S, P, T</td>
<td>25 HBB OTB</td>
</tr>
<tr>
<td>CPK (Li et al., 2022b)</td>
<td>HOG + CN + GCS + PM @ STC</td>
<td>APCE + MFD</td>
<td>Intel Xeon 10875H 2.3 GHz CPU</td>
<td>JL</td>
<td>C, S, P, T</td>
<td>105 HBB OTB</td>
</tr>
<tr>
<td>CF, MPMC (Liu et al., 2022)</td>
<td>DA + KCF</td>
<td></td>
<td></td>
<td>JL, SS</td>
<td>C, S, P, T</td>
<td>8 HBB OTB</td>
</tr>
<tr>
<td>SRN-TFM (Ruan et al., 2022)</td>
<td>DA + DM @ CRM</td>
<td>PSR + TFM</td>
<td></td>
<td>JL</td>
<td>C, S, P</td>
<td>74 HBB VOT</td>
</tr>
<tr>
<td>JSR Net (Song et al., 2022)</td>
<td>DA @ SiamRPN</td>
<td></td>
<td></td>
<td>JL, ISS</td>
<td>C</td>
<td>309 HBB OTB</td>
</tr>
</tbody>
</table>

(continued on next page)
<table>
<thead>
<tr>
<th>Tracker</th>
<th>Data source</th>
<th>Exploited features</th>
<th>GPU</th>
<th>AUC_S</th>
<th>Category</th>
<th>NoSO</th>
<th>Output</th>
<th>Benchmark</th>
<th>AUC_P (%</th>
<th>EAO A R FPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MBLT (Zhang et al., 2022)</td>
<td>Nvidia GPU</td>
<td>–</td>
<td>2.4 GHz CPU, 64 GB RAM, GTX 1080Ti</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td></td>
</tr>
<tr>
<td>ADNet (Nam et al., 2016)</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td></td>
</tr>
<tr>
<td>RT-2021 (Gao et al., 2019)</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td></td>
</tr>
<tr>
<td>GCT (Song et al., 2017)</td>
<td>MixFormer (Cui et al., 2024)</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td></td>
</tr>
</tbody>
</table>

For the Exploited features @ Tracker prototype, DA = deep appearance feature, PM = physical motion feature, MFD = background subtraction, MTF = multi-frame differencing, IS = interest point feature, CB = convolutional block, MTA = multi-task attention, MDC = multi-domain classification, TP = target proposal, PV = peak value, MTA = state aware indicator, NR = normal regression, RM = residual module, NN = neural network, MM = multi-model, MC = multi-channel, BA = bag of MVs, DMR = deep multi-resolution, L = lightweight, IU = inter-level, C = cross-level, AI = all-in-one, SI = self-improvement, N = network, GL = graph-based, MB = multi-body, DB = dual-body, GO = group of objects, MA = multi-agent, GPC = Gaussian process classifier, PV = peak value, TO = object-to-object, AIF = adaptive attention feature, CMA = center motion-aware, FCM = fully convolutional module, GCM = Gaussian convolution module, GNN = graph neural network, BNN = binary neural network, STM = spatio-temporal matching, DTT = dynamic temporal templates, DIL = dilated LSTM, LCT = local context transformer, VDM = vector data, GBO = Gaussian belief object, PRL = parameterized relation learning, CCF = cross-channel fusion, EAO = expected average overlap. For the PC (CPU, RAM, Nvidia GPU), CPU = Central Processing Unit, RAM = Random Access Memory, and GPU = Graphics Processing Unit. NoSO = (intersection over union) threshold. EAO = expected average overlap.
Trackers is essential to broaden the visual tracking community (Li et al., 2022c).

3. Review on satellite video trackers

Several trackers have been developed for SOT in SV, achieving superior results on their home-grown datasets. Table 2 outlines the characteristics of notable trackers. A comprehensive review and analysis of SV trackers is presented across six aspects including tracker prototype, exploited features, recognition and treatment of full occlusion (RTFO), rotation estimation, data source and tracked object, and evaluation benchmark.

3.1. Tracker prototype

Table 2 reveals that many trackers inherit the tracking paradigms of GV, such as DCF, SNN, CNN, and RNN. For example, some trackers (KCF_TFD (Du et al., 2018), HKCF (Shao et al., 2019b), VCF (Shao et al., 2019c), CFME (Xuan et al., 2020), CF_MFMC (Liu et al., 2021), RACF (Xuan et al., 2021), HMTS (Chen et al., 2022b), RANC (Chen et al., 2022c), and CF_MFMC (Liu et al., 2022)) are based on the KCF framework, while WTIC (Wang et al., 2020), CKFF (Guo et al., 2019), DF (Chen et al., 2022), and CPKF (Li et al., 2022b) are modeled on the CSK, DSST (Danellian et al., 2017), Staple, and STRCF, respectively, to achieve competitive speed. Some trackers, such as PASiam (Shao et al., 2019a), VAASN (Bi et al., 2021), and ThickSiam (Zhang et al., 2023), inherit the SiamFC, while HRSiam (Shao et al., 2021), ID-DSN (Zhu et al., 2021), and JSA Net (Song et al., 2022) inherit the SiamRPN. These SNN-based trackers strike a balance between tracking accuracy and speed. AD-OHNet (Cui et al., 2022b) is modeled on ADNet (Yun et al., 2018), which uses CNN to extract discriminative features of objects, while CRAM (Hu et al., 2020) uses convolutional regression networks to solve the regression problem and applies gradient descent in an end-to-end learning fashion. It is observed that most trackers inherit DCF and exploit hand-crafted features for SV object tracking, which may lead to unsatisfactory accuracy. In contrast, SNN, CNN, RNN, and Transformer-based trackers could emerge as the mainstream direction in SV tracking domain.

3.2. Exploited features

The features play a critical role in SV object tracking. Table 2 provides an overview of exploited features. These features can be broadly categorized into two types: spatial features and temporal features. Spatial features are primarily concerned with representing the appearance information by using both hand-crafted or deep features. Hand-crafted features such as HOG, color name, intensity, Gabor, CH, and local binary patterns (LBP) are commonly used to describe spatial texture and structure information. Deep appearance feature (DA) is one of the most common features, with the shallow layer containing low-level information with high spatial resolution, suitable for accurate localization. In contrast, the deep layer encodes high-level semantic information and is invariant to appropriate object changes. Hierarchical features of DA have been used in many trackers such as HRSiam (Shao et al., 2021), CRAM (Hu et al., 2020), ID-DSN (Zhu et al., 2021), and JSA Net (Song et al., 2022), with excellent results. Temporal features, on the other hand, focus on extracting inter-frame dynamic information using techniques such as multi-frame differencing (MFD), background subtraction (BS), optical flow (OF), deep motion feature (DM), and physical motion feature (PM). KCF_TFD (Du et al., 2018) fuses KCF and three-frame difference for tracking SV objects. Both PASiam (Shao et al., 2019a) and HRSiam (Shao et al., 2021) use BS features to assist with the tracking task. The OF is capable of representing inter-frame motion information and has been explored in many SV trackers such as MOFT (Du et al., 2019), HKCF (Shao et al., 2019b), VCF (Shao et al., 2019c), and RAMC (Chen et al., 2022c). Deep OF, as a common DM, has also yielded excellent tracking results in CRAM (Hu et al., 2020) and SRN-TFM (Ruan et al., 2022). Considering the relatively stable motion state of SV objects, the PM has been widely applied to most trackers such as CFFK (Guo et al., 2019), PASiam (Shao et al., 2019a), WTIC (Wang et al., 2020), CFM (Xuan et al., 2020), CF_MFMC (Liu et al., 2021), ID-DSN (Zhu et al., 2021), HMTS (Chen et al., 2022b), DF (Chen et al., 2022), STRCF_MIM (Li and Bian, 2022), CPKF (Li et al., 2022b), CF_MFMC (Liu et al., 2022), and SNN-TFM (Ruan et al., 2022). For the PM, some methods such as the Kalman filter (KF) (Kalman, 1960), motion smoothness (MS) (Wang et al., 2020), and motion trajectory averaging (MTA) (Xuan et al., 2020), are embedded to analyze the motion trajectory. Overall, combining the spatial feature with temporal feature can be an effective way to cope with challenging attributes, and its effectiveness has been validated.

3.3. Recognition and treatment of full occlusion

In SV, full occlusion (FO) is a common and challenging attribute due to the nadir view. To correctly track the object under FO, the tracker needs to solve three sub-problems (Xuan et al., 2020):

- Occlusion awareness: A tracker needs to be aware of the occurrence of object occlusion.
- Occlusion handling: A tracker is expected to overcome full occlusion without losing the object.
- End of occlusion awareness: A tracker needs to be aware of the end of the occlusion.

As shown in Table 2, existing trackers typically recognize and handle occlusions by comparing indicators with thresholds to solve the first and third sub-problems. Common indicators used in these trackers include average peak correlation energy (APCE) (Guo et al., 2019), peak value (PV) (Li et al., 2022b), tracking status monitoring indicator (TSMI) (Wang et al., 2020), peak to sidelobe ratio (PSR) (Ruan et al., 2022), and state aware indicator (SAI) (Chen et al., 2022). These metrics enable analysis of the tracking confidence and thus awareness of the FO. Experimental results demonstrate their effects. FO is usually accompanied by object disappearance. In this case, most trackers analyze historical motion information using traditional methods such as KF, MS, MTA, NR, interacting multiple model (IMM) (Li and Bian, 2022), and trajectory fitting motion (TFM) (Ruan et al., 2022) to predict the object state in the current frame. Additionally, some trackers use CNN methods such as fully convolutional network (FCN) (Zhang et al., 2022) and deep reinforcement learning (DRL) (Cui et al., 2022b) to analyze historical motion information and predict object states. While these methods have shown remarkable performance. However, FO remains a major challenge due to the diverse motion states of objects, such as straight-ahead movement, turning, lane changes, and overtaking. Therefore, more effective approaches are anticipated to address this issue in future studies.

3.4. Rotation estimation

Object rotation is a common phenomenon in SV, which can lead to accuracy degradation (Xuan et al., 2021). This problem has been successfully addressed by some excellent trackers, which can be categorized into two groups based on their outputs. Trackers with horizontal bounding box (HBB) outputs often experience scale changes due to object rotation. To solve this problem, some trackers such as WTIC (Wang et al., 2020), CF_MFMC (Liu et al., 2021), CF_MFMC (Liu et al., 2022), CPKF (Li et al., 2022b), DF (Chen et al., 2022), and SNN-TFM (Ruan et al., 2022) apply rotation invariance features to represent the tracked objects. For trackers that output OBB, a series of rotation patches with specific angle pools are listed to achieve a better match with the template. This approach allows the tracker to detect angle changes between adjacent frames and obtain a more accurate semantic representation.
RACF (Xuan et al., 2021) employs a similar strategy to address the rotation issue and proposes a method to estimate scale changes even with HBB outputs. In RAMC (Chen et al., 2022c), the rotation is decomposed into a translation solution to achieve adaptive rotation estimation of SV objects. Going forward, the spatial structure differences of sequence frames should be further explored to address unwanted rotation issues in SV tracking domain.

3.5. Data source and tracked object

Currently, video satellites are still in the developmental stage and are limited in number. As shown in Table 2, SVs are mainly provided by SS, JL, ISS, and Carbonite-2 (CB). Developed by Surrey Satellite Technology (SSTL), the CB delivers 1.2 m GSD RGB video and is capable of capturing video lasting approximately 120 s. Table 1 presents the detailed configurations and parameters of some video satellites. As shown in Table 2, most trackers use JL SVs due to their high quality. Tracked objects are mainly cars, ships, and planes because these objects are common and have a moderate aspect ratio. In contrast, train objects have a larger aspect ratio, which increases the difficulty of tracking. Therefore, the tracking of trains is very challenging and requires more attention in future research.

3.6. Evaluation benchmark

Table 2 shows that only a few trackers such as CFKF (Guo et al., 2019), CRAM (Hu et al., 2020), and SRN-TFM (Ruan et al., 2022) are evaluated via the Visual-Object-Tracking Challenge (VOT) benchmark (Kristan et al., 2018), whereas the others are evaluated via the one-pass evaluation (OPE) of the Object Tracking Benchmark (OTB) (Wu et al., 2013). The recent tracker REPS (https://github.com/YZCU/REPS) (Chen et al., 2024) also inherits the OTB. This is because the reset mechanism of the VOT may not be suitable for relatively short-term SV tracking tasks, especially in the case of frequent occlusions, dense objects, background clutters, and so on. In contrast, the OPE avoids the reset mechanism by initializing the tracker in the first frame and letting it estimate the object throughout the video. However, the OTB cannot accurately assess OBB results, and the precision score is susceptible to different objects.

4. Related datasets

Benchmark datasets are essential for the fair and standardized evaluation of trackers (Jiao et al., 2023). These benchmarks are typically categorized according to generic and specific applications (Fan et al., 2021). Table 3 provides details of some generic and specific benchmark datasets.

4.1. Generic benchmark datasets

Generic benchmark datasets usually contain a variety of objects gathered from natural scenes, such as vehicles, animals, balls, and human parts. As shown in Table 3, OTB50 (Wu et al., 2013) comprises 50 sequences with manually annotated HBBs for each frame. These videos include both color and gray sequences and are classified into 11 challenge attributes. Subsequently, OTB50 is extended to OTB100 (Wu et al., 2015), which consists of 100 videos with the same 11 attributes. NFS (Galoogahi et al., 2017a) is composed of 100 sequences with a frame rate of 240 FPS, which focuses on testing the impact of a high frame rate on tracking performance. Each sequence is labeled with nine attributes. VOT2018 (Kristan et al., 2018) contains short-term and long-term challenge splits. The VOT2018 Short-Term (VOT2018-ST) dataset...
comprises 60 sequences across 24 categories. The large-scale LaSOT consists of 1,120 training sequences and 280 testing sequences, which are annotated with HBB and categorized based on 14 attributes. TrackingNet (Muller et al., 2018) provides 60,643 sequences with 30,643 for training and 511 for testing, respectively. It has over 14 million HBB annotations covering 27 different object categories. While each sequence is represented by 15 attributes. GOT-10k (Huang et al., 2021) contains about 10,000 sequences, including 9,340 for training, 420 for testing, and 180 for validation. It populates 563 moving object categories, six attributes, and 87 motion patterns.

4.2. Specific benchmark datasets

Specific benchmark datasets are used to evaluate trackers under specific applications. As shown in Table 3, UAV123 (Mueller et al., 2016) contains 123 short sequences of nine object categories filmed by professional-grade UAVs, which are similar to SV sceneries. LasHeR (Li et al., 2022a) is a large-scale and high-diversity benchmark for RGB-thermal (RGBT) tracking. It consists of 1,224 pairs of visible and thermal infrared sequences of 32 object categories with over 730 K frames, and each sequence is annotated by 19 attributes. TOTB (Fan et al., 2021) offers 225 sequences aimed at diagnosing trackers under transparent objects. VISO (Yin et al., 2022) is a large-scale dataset with a wide range of HBB annotations for various SV tasks including moving object detection, SOT, and multiple object tracking. Among them, the SOT dataset offers 3,159 tracklets with about 1.12 M frames. SatSOT (Zhao et al., 2022) pays special attention to SOT in SVs and includes 105 sequences with 27,664 HBB annotations, 11 attributes, and four categories of typical objects (i.e., car, ship, plane, and train). SV248S (Li et al., 2022c) provides 249 objects from six SVs captured by JL, with 10 attributes and three categories of objects (i.e., car, plane, and ship). It uses the tight polygon to label the object, which is particularly effective in representing plane objects with relatively complex contours. XDU-BDSTU (Zhang et al., 2022) contains 11 attributes and 20 objects from nine JL SVs, which is specially designed for vehicle tracking in SV. The object is labeled with HBB. ThickSiam_D (Zhang et al., 2023) includes 12 objects derived from eight SVs with a total of 5.55 K frames with HBB annotations. In addition, SAT-MTB (Li et al., 2023), a recent multi-task benchmark dataset, has been proposed for SV object detection, tracking, and segmentation. The proposed OOTB consists of 110 sequences covering typical object categories, such as car, ship, plane, and train, with 12 challenging attributes and a total of 29,890 frames. It is a specific dataset tailored for SOT in SV and includes a small portion of the data from (He et al., 2022; Yin et al., 2022; Zhao et al., 2022). Notably, it is the first benchmark to apply fine OBB annotations to ensure the accuracy of object scale, center, orientation, and motion direction as much as possible.

5. OOTB

5.1. Multi-platform data collection

Currently, available SVs are limited. Datasets collected by a single platform have facilitated the development of SOT. However, this can result in similar characteristics in terms of the spatial resolution, frame rate, and spectral features, thus limiting the diversity of the SV dataset. Towards this end, the OOTB dataset is sampled from multiple satellite platforms, such as JL, SS, and ISS. Moreover, we have also included part of the SV data from (He et al., 2022; Yin et al., 2022; Zhao et al., 2022). The multi-platform data would satisfy the need for dataset diversity and allow for better representation and generalization.

5.2. High-quality annotation with OBBs

A tracking dataset should be equipped with high-quality annotations. In SV, objects are typically shown in rectangles with orientation, making OBB the preferred format. Compared to HBB used in (He et al., 2022; Yin et al., 2022; Zig, et al., 2022; Zhang et al., 2023; Zhao et al., 2022), OBB provides more accurate representations, such as position, size, and orientation. Additionally, OBB is more compact and less susceptible to background interferences, especially for objects with large aspect ratios and angles, as shown in Fig. 4. Therefore, we aim to represent objects in a compatible metadata format by using OBBs, which can be easily transformed into the HBB format through batch procedures.

Specifically, the OBB description includes coordinates of the four corners. We use the roLabelImg software and zoom in 10 times for ac-

![Fig. 4. Visualization of the HBB (red) and OBB (green). (a) Ship. (b) Train. Compared to HBB, OBB is more compact and suppresses the interference from the background, especially for objects with large aspect ratios and angles. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)](image-url)
curate annotation. The annotation example of a train is shown in Fig. 5. The labeling format of roLabelImg is $(x, y, w, h, \theta)$, where $(x, y)$ represents the center, and $w$, $h$, and $\theta$ represent the width, height, and rotation angle of the bounding box, respectively. To conform to the generic description, we transform the annotation format into float type with four decimal places for precision representation. To ensure annotation consistency, we impose annotation consistency constraints on the total annotation process, as shown in Fig. 6. In the evaluation process, we do not directly consider the direction angles, rather preferring to reflect the direction angle deviation of trackers to the metrics of the evaluation protocol. During tracking, the tracker usually combines the foreground and background information for tracking rather than just relying on that of the object itself (Javed et al., 2022). In the satellite video, the object movement is slow, and the angle change of adjacent frames is relatively slight due to the high frame rates (e.g., 25 FPS) and the long-distance satellite platform. Therefore, it is difficult for trackers to yield a large angle deviation. Each sequence is cropped and labeled by the same person to ensure a uniform annotation protocol and an expert would check and fine-tune the OBB if necessary. Each sequence is refined by at least three people. With these supervision strategies, we can guarantee high-quality OBB annotations.

### 5.3. Data statistics

Benefiting from the multi-platform data, we could collect prosperous SVs with different regions, spectral features, spatial resolutions, and attributes. In the following, we present detailed statistics and analysis of the OOTB.

#### 5.3.1. Scenery type

The complexity of the scenarios contributes to the diversity of the dataset. As shown in Fig. 7, satellites can observe large areas and also produce dynamic and diverse scenes, which pose a great challenge to trackers. On the one hand, different categories of objects are usually captured in different scenes. For instance, in the first row of Fig. 7, a car drives on a crowded road, a ship sails on the sea, a plane parks at an
Fig. 7. Visualization of the scenery diversity. Each row shows four object categories, while each column shows the same category in different scenarios.

Fig. 8. Overview of the OOTB dataset. It has four object categories including 45 cars, 30 ships, 25 planes, and 10 trains. The blue bar represents the size of the object. The red line indicates the number of frames. The average sizes are 109.7, 238.7, 2075.3, and 1949.0 pixels, respectively. The average frame length of the dataset is 271.7. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
airport, and a train moves towards the city. On the other hand, similar objects may appear in distinct scenarios. As shown in the third column of Fig. 7, the plane flies over diverse backgrounds and has different motion characteristics (e.g., takeoff, cruising, and landing). In particular, the diversity of satellite platforms results in differences in video spatial–temporal resolution, spectral range, and imaging views. This can help test the robustness of trackers and facilitate designing a reasonable tracking scheme.

5.3.2. Specific category

As illustrated in Fig. 8, the OOTB dataset consists of 110 sequences including 45 cars, 30 ships, 25 planes, and 10 trains, totaling 29,890 frames. The average, shortest, and longest videos contain 271.7, 90, and 750 frames, respectively. Car and train are the most challenging categories, while ship and plane are relatively easier to track. This is because cars are typically smaller in size and have more complex backgrounds, while trains have larger aspect ratios and more frequent non-rigid deformations. Consequently, more cars are included in the OOTB, while trains are relatively less common due to their infrequent appearance in typical scenes.

5.3.3. Size and aspect ratio

Object size provides two essential pieces of information: 1) it helps determine the search region of the tracker so that computational resources can be allocated appropriately, and 2) it serves as a measure of tracking difficulty, with smaller objects being more challenging to track. Fig. 8 illustrates the object area distribution of the OOTB. The average object areas for cars, ships, planes, and trains are 109.7, 238.7, 2075.3, and 1949.0 pixels, respectively. More than 75% of the sequences have an object area smaller than 1113 pixels. As complementary information, the aspect ratio can finely describe the shape of an object. It reflects the relationship between the aspect ratio and area. Fig. 9 shows the Gaussian kernel density estimate for the object aspect ratio. The average aspect ratios for cars, ships, planes, and trains are 2.0, 2.0, 1.2, and 1.0, respectively, and their maximum aspect ratios are 4.7, 3.4, 1.6, and 14.8, respectively. The larger the aspect ratio, the more likely to encounter background interference and rotation issues, especially for trackers that only output HBB.

5.4. Attribute

GV is usually captured by a variety of optical or infrared devices such as handheld cameras, mobile surveillance devices, UAVs, and infrared cameras, which causes multiple challenges (e.g., fast motion, out-of-view, aspect ratio change, and thermal crossover). However, SV is significantly different from GV in terms of imaging devices, observation means, imaging regions, atmospheric environment, etc. The main challenges are summarized below.

![Figure 9. The aspect ratios of the OOTB dataset. The average aspect ratio of the car and ship is 2.0, while the average aspect ratio of the plane is 1.2. The train has a aspect ratio ranging from 4.4 to 14.8. Rag denotes the range of the aspect ratio. Avg denotes the average aspect ratio.](image-url)
The diagonal data corresponds to the distribution in the overall dataset, and each row or column represents the distribution of the attribute subset.

5. Evaluation protocol

A high-precision evaluation protocol is proposed for fair comparisons. Firstly, we perform the OPE evaluation and obtain the precision plot of all trackers. In addition, the normalized precision plot is used for evaluation to avoid the effect of object size. Finally, the success plot is included, and an FPS metric is used to measure the tracking speed.

5.5. Precision plot

The precision plot records the percentage of frames where the center location error (CLE) is smaller than the predefined threshold. As shown in Fig. 11, the CLE is determined by computing the average Euclidean distance between the center of the predicted bounding box (i.e., HBB or OBB) $(x, y)$ and the ground truth (i.e., OBB) $(X, Y)$. Let $d_c$ denote the CLE, defined as

$$d_c = \sqrt{(x - X)^2 \text{+} (y - Y)^2}$$  \hspace{1cm} (1)

To account for the low spatial resolution of SVs and the small size of objects, we use the threshold varied from 1 to 30 for the precision plot and measure the tracking performance by using the precision rate (PR) at a threshold of 5 pixels. This contrasts with other benchmarks such as UAV123 (Mueller et al., 2016), OTB (Wu et al., 2013, 2015), TrackingNet (Muller et al., 2018), and LaSOT, which use a threshold varied from 1 to 50 and a threshold of 20 pixels. The reason for this difference is that SV objects are typically smaller than GV objects, and a tracking drift of 5 pixels in SV (i.e., Fig. 12(a)) is almost equivalent to that of 20 pixels in GV (i.e., Fig. 12(b)). Notably, the precision plot is sensitive to the image resolution and object size (Muller et al., 2018). As shown in Fig. 12(a) and Fig. 12(c), their resolution is the same. However, the drift magnitudes of the former are more significant than that of the latter.

5.5.2. Normalized precision plot

To compensate for the precision plot, we further propose the normalized precision plot (Muller et al., 2018) for evaluating SV trackers. The normalized precision plot shows the percentage of frames for which the normalized CLE is smaller than the predefined threshold varied from 0 to 1. Let $d_a$ denote the normalized CLE, defined as

$$d_a = \left(\frac{(x - X)}{W}\right)^2 \text{+} \left(\frac{(y - Y)}{H}\right)^2$$  \hspace{1cm} (2)

where $W$ and $H$ are the width and height of the ground truth.

Considering that most trackers can only produce HBB, computing the normalized CLE directly using the ground truth (i.e., OBB) and the predicted result (i.e., HBB) may lead to inconsistent representations. This is because there is no explicit correspondence between their widths and heights. To this end, we propose a strategy for adaptively solving the HBB and OBB format. Subsequently, its width and length are applied for evaluation. The strategy is also embedded in the initialization process for trackers that can only receive the HBB format. For trackers that can predict the OBB format, we naturally employ the height and width of the OBB to compute the normalized CLE. In OOTB, we use the area under the curve (AUC) of the normalized precision plot, i.e., normalized pre-
cision rate (NPR), to rank trackers and avoid unfair comparisons due to specific thresholds.

5.5.3. Success plot

In the success plot, the success rate (SR) aims to calculate the percentage of successful frames where the overlap surpasses the threshold varied from 0 to 1. Given the predicted bounding box $r_p$ and ground truth $r_g$, the overlap score $s$ is obtained by

$$s = \frac{|r_p \cap r_g|}{|r_p \cup r_g|}$$ (2)

where $\cap$ and $\cup$ represent intersection and union, respectively, and $|\cdot|$ denotes the number of pixels in the given region.

As discussed above, OOTB is annotated with OBB format, which is different from previous SV datasets annotated with HBB format, such as VISO (Yin et al., 2022), SatSOT (Zhao et al., 2022), XDU-BDSTU (Zhang et al., 2022), ThickSiam_D (Zhang et al., 2023), and AIR-MOT (He et al., 2022). For fair assessment, two methods for resolving the overlap score are proposed. The first method is designed to assess the tracker with HBB output. Concretely, the ground truth (i.e., OBB) and predicted bounding box (i.e., HBB) are converted into corresponding external HBBs. The intersection and union regions of these two HBBs are then obtained to calculate the overlap, as shown in Fig. 13(a). While the second method is designed to assess the tracker with OBB output. In particular, we directly

---

*Fig. 11. Visualization of the CLE. Ground truth and predicted result are marked with green and red boxes, respectively. (a) and (b) show the CLE with HBB and OBB predictions, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)*

*Fig. 12. Comparison of tracking drift for different videos and objects. Drift pixels are displayed in the upper left corner. Ground truth and predicted result are marked with green and red boxes, respectively. (a) and (c) show vehicle and aircraft objects from SVs, respectively. (b) shows an object from the GV. (a) and (b) suffer from approximate drift magnitudes, even though (a) drifts by 5 pixels while (b) drifts by 20 pixels. Both (a) and (c) drift by 5 pixels, but their drift magnitudes vary greatly due to the size of the object. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)*
compute the intersection and union regions between the ground truth (i.e., OBB) and predicted bounding box (i.e., OBB), as shown in Fig. 13(b). The AUC of the success plot (i.e., SR) is used to rank trackers.

Overall, the larger the PR, NPR, SR, and FPS values, the better the tracking performance.

5.6. Selecting 33 SOTA trackers for evaluation

In OOTB, we compare and analyze 33 representative SOTA trackers with a total of 58 models covering different features, backbones, and tracker tags. The compared algorithms are CSK (Henriques et al., 2012), SAMF (Li and Zhu, 2015), DAT (Possegger et al., 2015), KCF (Henriques et al., 2015), SRDCF, Staple, DSST, BACF (Galoogahi et al., 2017b), SiamRPN (Li et al., 2018a), DaSiamRPN (Zhu et al., 2018), ARCF (Huang et al., 2019), SiamRPN++ (Li et al., 2019a), UpdateNet (Zhang et al., 2019), SiamDW (Zhang, 2019b), SiamMask (Wang et al., 2019c), SiamCAR (Chen et al., 2020), SiamFC++ (Xu et al., 2020), AutoTrack (Li et al., 2020), CFME (Xuan et al., 2020), SiamGAT (Guo et al., 2021), LightTrack (Yan et al., 2021), Stark (Yan et al., 2021a), SiamCAR (Cui et al., 2022a), OSTrack (Ye et al., 2022a), SimTrack (Chen et al., 2022a), DF (https://github.com/YZCU/DF) (Chen et al., 2022), RAMC (Chen et al., 2022c), SBT (Xie et al., 2022), GRM (Gao et al., 2023), SeqTrack (Chen et al., 2023a), ARTrack (Wei et al., 2023), ODTrack (Zheng et al., 2024), and SMAT (Yelluru Gopal and Amer, 2024). These SOTAs cover all mainstream tracking paradigms. Usually, a tracker is trained using distinctive models and datasets to meet diverse needs. To explore suitable methods and models for SOT in SV, we benchmark 33 SOTAs with a total of 58 models.

6. Experiments and analysis

In this section, we perform comprehensive experiments and analysis. Section 6.1 quantitatively compares all trackers including overall, category-based, and attribute-based evaluations. Section 6.2 presents the qualitative results. While Section 6.3 compares and analyzes the running speeds.

6.1. Quantitative evaluations

6.1.1. Overall evaluation results

Here we provide a comprehensive assessment on OOTB. To be fair, we conduct experiments using the officially provided models and corresponding configuration parameters.

Table 6 shows the characteristics, overall results, category-based results, and running speed for all 33 trackers with a total of 58 models. The benchmark includes four metrics (i.e., PR, NPR, SR, and FPS). Fig. 14 displays the precision plot, normalized precision plot, and success plot for the top 30 trackers. The values in the legend denote PR, NPR, and SR, respectively. SiamCAR with default parameters attains outstanding performance in both precision plot and normalized precision plot with a PR of 0.824 and NPR of 0.779. Additionally, SiamFC++ with AlexNet adopts the anchor-free idea and proposes a set of object state estimation guidelines, which secures top ranking in the SV tracking.

In the success plot, SiamDW with CIRNext22 performs remarkably well with an SR of 0.645, surpassing the fourth-place DF by 2.2%. RAMC can combine the appearance and motion features to deal with object rotation and tracking drifts. It achieves a satisfactory SR of 0.598. Since RAMC can estimate compact OBB, it is prone to tracking position bias. Therefore, the results obtained in the precision map and normalized precision map are relatively poor. Furthermore, DSST produces satisfactory results in PR, NPR, and SR scores. Notably, SiamDW including SiamDW_CIRNext22, SiamDW_CIRIncep22, and SiamDW_CIRResNet22, achieves competitive PR, NPR, and SR scores. This indicates that the residual modules and deep-wide network structure of SiamDW aid in mining the semantic information of SV objects.

6.1.2. Category-based evaluation results

In this section, we perform a category-based evaluation on OOTB. As shown in Fig. 8, SV objects can typically be divided into four categories. Table 6 summarizes the characteristics and results of the category-based evaluation. Fig. 15 shows the precision plot, normalized precision plot, and success plot for the top 30 trackers. It is observed that the plane is the easiest object to be tracked. The top 30 trackers exhibit tighter tracking curves and higher PR, NPR, and SR scores. This is attributed to the large size and the significant texture and structural information,
<table>
<thead>
<tr>
<th>Tracker</th>
<th>Venue</th>
<th>Feature/Backbone</th>
<th>Tracker tag</th>
<th>Specific tracker name</th>
<th>FPS</th>
<th>Overall Result</th>
<th>Category-based result</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSK</td>
<td>ECCV 2012</td>
<td></td>
<td>CSK</td>
<td>Default</td>
<td>229</td>
<td>465.8</td>
<td>0.483 0.471 0.359</td>
</tr>
<tr>
<td>SAMF</td>
<td>ECCV 2015</td>
<td>HOG + CN + I</td>
<td>SAMF</td>
<td>Default</td>
<td>229</td>
<td>0.450 0.578 0.478</td>
<td>0.483 0.471 0.359</td>
</tr>
<tr>
<td>DAT</td>
<td>CVPR 2015</td>
<td>CH</td>
<td>DAT</td>
<td>Default</td>
<td>229</td>
<td>0.590 0.589 0.459</td>
<td>0.614 0.555 0.407</td>
</tr>
<tr>
<td>KCF</td>
<td>TAMPI 2015</td>
<td>HOG</td>
<td>KCF</td>
<td>Default</td>
<td>229</td>
<td>0.474 0.569 0.453</td>
<td>0.519 0.472 0.376</td>
</tr>
<tr>
<td>SRDCF</td>
<td>ICCV 2015</td>
<td>HOG</td>
<td>SRDCF</td>
<td>Default</td>
<td>229</td>
<td>0.493 0.529 0.428</td>
<td>0.486 0.454 0.343</td>
</tr>
<tr>
<td>Staple</td>
<td>CVPR 2016</td>
<td>HOG + CN</td>
<td>Staple</td>
<td>Default</td>
<td>229</td>
<td>0.772 0.730 0.607</td>
<td>0.799 0.739 0.649</td>
</tr>
<tr>
<td>DSST</td>
<td>TAMPI 2015 + I</td>
<td></td>
<td>DSST</td>
<td>Default</td>
<td>229</td>
<td>0.313 0.800 0.767</td>
<td>0.895 0.787 0.609</td>
</tr>
<tr>
<td>BACF</td>
<td>ICCV 2017</td>
<td>HOG</td>
<td>BACF</td>
<td>Default</td>
<td>229</td>
<td>0.795 0.750 0.547</td>
<td>0.847 0.750 0.499</td>
</tr>
<tr>
<td>SiamRPN</td>
<td>CVPR 2018</td>
<td></td>
<td>SiamRPN</td>
<td>Default</td>
<td>229</td>
<td>0.605 0.624 0.464</td>
<td>0.616 0.544 0.442</td>
</tr>
<tr>
<td>DaSiamRPN</td>
<td>ECCV 2018</td>
<td></td>
<td>DaSiamRPN</td>
<td>Default</td>
<td>229</td>
<td>0.803 0.811 0.648</td>
<td>0.977 0.778 0.613</td>
</tr>
<tr>
<td>ARCF</td>
<td>ECCV 2019</td>
<td>HOG + CN</td>
<td>ARCF</td>
<td>Default</td>
<td>229</td>
<td>0.751 0.735 0.603</td>
<td>0.803 0.713 0.558</td>
</tr>
<tr>
<td>SiamRPN++</td>
<td>CVPR 2019</td>
<td></td>
<td>SiamRPN++</td>
<td>Default</td>
<td>229</td>
<td>0.789 0.770 0.595</td>
<td>0.820 0.727 0.552</td>
</tr>
<tr>
<td>UpdateNet</td>
<td>ICCV 2019</td>
<td></td>
<td>UpdateNet</td>
<td>Default</td>
<td>229</td>
<td>0.641 0.708 0.550</td>
<td>0.853 0.752 0.605</td>
</tr>
<tr>
<td>AutoTrack</td>
<td>CVPR 2020</td>
<td>HOG + CN + I</td>
<td>AutoTrack</td>
<td>Default</td>
<td>229</td>
<td>0.702 0.728 0.550</td>
<td>0.716 0.650 0.517</td>
</tr>
<tr>
<td>CFME</td>
<td>TGRS 2020</td>
<td>HOG</td>
<td>CFME</td>
<td>Default</td>
<td>229</td>
<td>0.753 0.714 0.610</td>
<td>0.831 0.717 0.517</td>
</tr>
<tr>
<td>SiamGAT</td>
<td>CVPR 2020</td>
<td></td>
<td>SiamGAT</td>
<td>Default</td>
<td>229</td>
<td>0.580 0.534 0.463</td>
<td>0.530 0.460 0.357</td>
</tr>
<tr>
<td>LightTrack</td>
<td>ICCV 2021</td>
<td>Mobile</td>
<td>LightTrack</td>
<td>Default</td>
<td>229</td>
<td>0.611 0.687 0.530</td>
<td>0.542 0.482 0.372</td>
</tr>
<tr>
<td>Starday</td>
<td>ECCV 2021</td>
<td></td>
<td>Starday</td>
<td>Default</td>
<td>229</td>
<td>0.507 0.513 0.451</td>
<td>0.418 0.409 0.352</td>
</tr>
<tr>
<td>SimTrack</td>
<td>ECCV 2022</td>
<td></td>
<td>SimTrack</td>
<td>Default</td>
<td>229</td>
<td>0.535 0.544 0.430</td>
<td>0.487 0.432 0.364</td>
</tr>
<tr>
<td>DF</td>
<td>JSTARS 2022</td>
<td></td>
<td>DF</td>
<td>Default</td>
<td>229</td>
<td>0.754 0.747 0.623</td>
<td>0.784 0.794 0.642</td>
</tr>
<tr>
<td>RAMC</td>
<td>RS 2022</td>
<td>HOG + OF</td>
<td>RAMC</td>
<td>Default</td>
<td>229</td>
<td>0.781 0.703 0.598</td>
<td>0.747 0.624 0.507</td>
</tr>
<tr>
<td>SBT</td>
<td>CVPR 2018</td>
<td></td>
<td>SBT</td>
<td>Default</td>
<td>229</td>
<td>0.700 0.682 0.525</td>
<td>0.489 0.373 0.273</td>
</tr>
<tr>
<td>GRM</td>
<td>CVPR 2018</td>
<td></td>
<td>GRM</td>
<td>Default</td>
<td>229</td>
<td>0.645 0.629 0.500</td>
<td>0.503 0.448 0.355</td>
</tr>
<tr>
<td>SeqTracker</td>
<td>CVPR 2020</td>
<td></td>
<td>SeqTracker</td>
<td>Default</td>
<td>229</td>
<td>0.600 0.624 0.464</td>
<td>0.498 0.446 0.348</td>
</tr>
<tr>
<td>ARTrack</td>
<td>CVPR 2020</td>
<td></td>
<td>ARTrack</td>
<td>Default</td>
<td>229</td>
<td>0.732 0.731 0.649</td>
<td>0.723 0.709 0.512</td>
</tr>
<tr>
<td>ODTrack</td>
<td>AAIC 2020</td>
<td></td>
<td>ODTrack</td>
<td>Default</td>
<td>229</td>
<td>0.686 0.696 0.526</td>
<td>0.680 0.674 0.505</td>
</tr>
<tr>
<td>SMAT</td>
<td>WACV 2024</td>
<td></td>
<td>SMAT</td>
<td>Default</td>
<td>229</td>
<td>0.708 0.715 0.534</td>
<td>0.665 0.587 0.445</td>
</tr>
</tbody>
</table>

**CPU** | **GPU** | **NPR** | **SR** | **CPU** | **GPU** | **NPR** | **SR** | **CPU** | **GPU** | **NPR** | **SR** | **CPU** | **GPU** | **NPR** | **SR** | **CPU** | **GPU** | **NPR** | **SR** | **CPU** | **GPU** | **NPR** | **SR** | **CPU** | **GPU** | **NPR** | **SR** | **CPU** | **GPU** | **NPR** | **SR** | **CPU** | **GPU** | **NPR** | **SR** | **CPU** | **GPU** | **NPR** | **SR** | **CPU** | **GPU** | **NPR** | **SR** | **CPU** | **GPU** | **NPR** | **SR** | **CPU** | **GPU** | **NPR** | **SR** | **CPU** | **GPU** | **NPR** | **SR** | **CPU** | **GPU** | **NPR** | **SR** | **CPU** | **GPU** | **NPR** | **SR** | **CPU** | **GPU** | **NPR** | **SR** | **CPU** | **GPU** | **NPR** | **SR** |**
which facilitates the extraction of discriminative features. In this case, RAMC, DF, and CFME, three tailored trackers for SV tracking, perform well, ranking 5th, 7th, and 9th in the success plot, respectively. This is because of their ability to combine appearance and motion features to mitigate tracking drifts. Ship objects have relatively large sizes and stable motion patterns. However, the interference of tail waves and complex backgrounds may cause tracking drifts. Car objects face various challenges due to their small size, complex motion patterns, and similar contexts. SOTA trackers, such as DF, Staple, UpdateNet, SiamFC+++, and SiamCAR, can adapt to object changes and produce acceptable results. However, compared to plane and ship objects, the small size of cars leads to lower PR, NPR, and SR scores for almost all trackers. For the train object, it is one of the most challenging categories, and almost all SOTAs exhibit inferior performance. This is mainly due to the larger aspect ratios and more frequent non-rigid deformations compared to other categories. In this case, almost all the top trackers such as SeqTrack, ODTtrack, GRM, and ODTtrack use CNN features, as shown in Fig. 15(l). In contrast, hand-crafted feature-based trackers such as CSK, DSST, and SAMF have difficulty adapting to complex scenarios due to limited representations. While most trackers achieve the worst results in the train category, indicating that more attention should be paid to addressing this issue.

6.1.3. Attribute-based evaluation results

To evaluate the strengths and limitations of trackers, we perform attribute-based evaluation. Table 7 shows the SR scores for each attribute, and Fig. 16 presents the success plot for the top 30 trackers. DF and CFME achieve significant results in terms of PO, FO, IV, MB, BC, LT, IM, and AM, due to their ability to sense the tracking confidence of current frames and predict the object position in subsequent frames. The most prevalent challenge is IPR, where RAMC achieves the optimal SR score of 0.601, which is 1.7 % higher than the second-best SiamDW (SiamDW_CIRIncep22). SiamRPN++ (SiamRPN++_Def) obtains an SR score of 0.577 and ranks 3rd. Besides RAMC, the best three correlation filter-based trackers are DSST, DF, and Staple, with SR scores of 0.531, 0.524, and 0.522, respectively. Regarding the FO attribute, CFME ranks first with an SR score of 0.490, followed by SiamDW (SiamDW_CIResNet22), SiamDW (SiamDW_CIRNext22), and DF. These SOTAs encounter a severe drop in SR score, which indicates that FO is extremely challenging in SV. Fig. 16 also demonstrates that deep learning-based trackers are more robust in handling challenging attributes in the SV tracking domain. Whereas the correlation filter-based trackers, except for DF, DSST, and CFME, are relatively weak in addressing these challenges. Table 8 presents the NPR scores of all trackers on 12 attributes, and Fig. 17 presents the normalized precision plot for the top 30 trackers. It is found that the NPR scores also show a severe decrease in FO, confirming that FO is extremely challenging in SV tracking.

6.2. Qualitative evaluations

For qualitative evaluation, we visualize nine SOTA trackers including ODTTrack (ODTrack_L), SeqTrack (SeqTrack_Def), ARTrack, CFME, SiamFC++ (SiamFC++_Alex), DSST, SiamDW (SiamDW_CIRNext22), RAMC, and DF, covering a wide range of tracking paradigms. To showcase our performance, we have selected six sequences with diverse attributes and object categories, namely Car_16, Ship_10, Plane_2, Plane_21, Train_1, and Train_5. Fig. 18 shows the qualitative results to help intuitively understand the tracking performance. In the Car_16 sequence, the object undergoes in-plane rotation, full occlusion, and illumination variation, which challenge the tracking algorithms. The results show that SOTA trackers encounter tracking failure when the car is completely occluded. Compared to other trackers, DF and DSST perform better and cross the obstacle. CFME is also capable of overcoming occlusion but may fail in the case of long-duration occlusion. The Train_1 sequence, as one of the most challenging sequences,
encounters deformation, in-plane rotation, illumination variation, motion blur, and out-of-normal. In particular, the train object tends to undergo non-rigid deformation, making it extremely difficult to track accurately. In this case, DSST, DF, and CFME initially drift away from the object (e.g., frame #0164). While RAMC, ARTrack, and SiamFC++ (SiamFC++_Alex) can adapt to object changes and achieve better tracking performance. More visual samples can be found in Fig. 18.

Based on qualitative results, we can draw several conclusions. Firstly, combining appearance with motion features is useful in handling multiple challenges. Secondly, deep feature-based trackers are more robust in the SV tracking domain than hand-crafted feature-based trackers. Thirdly, tracking the train object is extremely demanding and requires more attention. Finally, the SOT of SV is far from being well resolved. There is still room for improvement in this field.

Fig. 15. The precision plot (column 1), normalized precision plot (column 2), and success plot (column 3) for the top 30 trackers. Rows 1 to 4 show the results for car, ship, plane, and train, respectively.
6.3. Running speed analysis

Table 7 presents the FPS metric on Central Processing Unit (CPU) and Graphics Processing Unit (GPU) devices. Traditional SOTA trackers, such as CSK, SAMF, DAT, Staple, and AutoTrack, mainly utilize the CPU device. For trackers using deep features, they usually rely on the GPU device and are tested on the NVIDIA GeForce RTX 4060 GPU. CSK achieves the fastest speed of 257.2 FPS on GPU. DaSiamRPN (DaSiamRPN_OTB) follows with 168.9 FPS. In contrast, recent SOTAs achieve relatively low tracking speeds, such as SMAT, ODTrack, ARTrack, SeqTrack, GRM, and SBF. Therefore, realizing the accuracy-speed trade-off is subject to further research in the field of satellite video object tracking.

7. Discussion and recommendations for future work

Remote sensing Earth observation techniques have achieved vigorous development in change detection (Wang et al., 2022c), anomaly detection (Cheng et al., 2024; Lin et al., 2023), clustering (Guan et al., 2023), segmentation (Wang et al., 2022b), etc. However, previous research has mainly focused on image data. The emergence of video satellites has opened up a new era of remote sensing Earth observation from static images to real-time videos. SOT in SV is one of
the most fundamental tasks in the intelligent interpretation of remote sensing, holding potential applications in traffic sensing and modeling, wildfire suppression, sustainable fisheries (Shao et al., 2021), etc. Although some progress has been made, there are still issues that impede development. In this article, firstly, we review various tracking paradigms and frameworks that cover both general video and satellite video domains to discover potential prospects in the SV tracking domain.

Secondly, we develop the first available oriented object tracking benchmark OOTB for SOT in SV. OOTB benchmarks various SOTAs and aims to identify the strengths of trackers and explore the intrinsic factors that contribute to effective tracking. Through comprehensive comparison and analysis, we have summarized several thoughts on how to facilitate SV object tracking.
properties similar to those of surrounding objects, but in opposite motion directions. One such scenario is anisotropic motion (i.e., AM), where an object moves with similar amplitude to surrounding objects, but in the opposite direction. In this case, relying on appearance information is difficult to distinguish it from the surrounding objects. By leveraging optical flow information, it is possible to discriminate the objects from the background based on their relative motion (Hu et al., 2020). Additionally, incorporating object trajectory information can further improve tracking performance in challenging scenarios. By analyzing the historical trajectory, it is able to predict future locations and adjust accordingly (Yang et al., 2023).

7.1. Synergy of appearance information and motion cues

Combining appearance information and motion cues can effectively handle challenging tracking scenarios. In particular, the use of optical flow and historical trajectory is useful when objects exhibit motion properties similar to those of surrounding objects, but in opposite motion directions. One such scenario is anisotropic motion (i.e., AM), where an object moves with similar amplitude to surrounding objects, but in the opposite direction. In this case, relying on appearance information alone may not be sufficient to achieve accurate tracking. This is because there is little difference in the object’s appearance. Therefore, it is difficult to distinguish it from the surrounding objects. By leveraging optical flow information, it is possible to discriminate the objects from the background based on their relative motion (Hu et al., 2020). Additionally, incorporating object trajectory information can further improve tracking performance in challenging scenarios. By analyzing the historical trajectory, it is able to predict future locations and adjust accordingly (Yang et al., 2023).

7.2. Dense object

The dense object is a significant challenge in the SV tracking domain. It is necessary to exploit more discriminative features to accurately identify track objects (Song et al., 2022). One approach is to analyze the spatial distribution of the tracked object and surrounding objects. In this way, the tracker can identify and track the object of interest, even
when multiple objects are close. To accomplish this, the tracker should extract and analyze highly discriminative object features. In addition, a tracker is expected to possess the ability to detect the object’s motion in its vicinity. This involves analyzing the motion patterns of nearby objects, as well as detecting any changes in their motion states. In summary, achieving accurate tracking under dense objects requires the tracker to exploit more discriminative features and detect the motion state of nearby objects.

7.3. Motion estimation

The non-stationary background is an issue for SOT in SV. With the high-speed moving platform and nadir view, trackers need to eliminate background motion and focus on the actual motion. Additionally, by
integrating multi-modal data such as Global Navigation Satellite System data (Zhou et al., 2021), high-resolution optical images (Guan et al., 2022), radar data (Garnot et al., 2022), and synthetic aperture radar data (Peng et al., 2023), we would capture precise geographic coordinates and object trajectories. As a result, the tracker can get a more complete picture of the object to predict its future motion and adjust accordingly.

7.4. Precise object representations

Most trackers can only generate HBB results that lack important semantic information such as orientation and shape. This can lead to performance degradation, particularly for SV objects with non-rigid deformation. Therefore, a superior tracker is expected to yield an accurate representation of the object, such as center position, scale, orientation, and shape (Chen et al., 2024). By leveraging these additional cues, it is possible to achieve more accurate and robust tracking performance.

7.5. Suitable backbones and features

In the SV tracking domain, the backbone and training dataset are typically borrowed from GV. Considering the significant differences between the SV and GV domains, it is urgent to develop the backbones and features suitable for SV object tracking. Moreover, pre-training using massive remote sensing datasets may result in substantial performance improvements.

7.6. Video enhancement

Video enhancement is a viable option for improving tracking performance. The tracker can benefit from enhancement processes such as video adjustment and reconstruction (Wang et al., 2023). In addition, the space–time super-resolution (Xiao et al., 2022) enables to obtain fine-grained features in both spatial and temporal dimensions, which provides rich motion cues and minimizes the risk of tracking drifts.
8. Conclusions

In this article, we first present a systematic review of tracking methods and datasets followed by introducing the proposed oriented object tracking benchmark OOTB. It is the first publicly available benchmark with high-quality oriented bounding box annotations and high-precision evaluation protocols in the satellite video single object tracking domain. OOTB includes 110 sequences captured by multiple satellite constellations, with a total of 29,890 frames, covering diverse object categories. To ensure comprehensive and fair evaluation, a protocol is proposed. We also benchmark 33 SOTA trackers including 58 models with different features, backbones, and tracker tags on OOTB. Extensive experiments and analysis are conducted in terms of the overall, category-based, and attribute-based results. Furthermore, the qualitative evaluation and speed analysis demonstrate that satellite video object tracking remains a challenging and far-from being resolved. Finally, several thoughts on facilitating satellite video tracking tasks are summarized. It is believed that this work will spark interest in satellite video tracking, which in turn will lead to advances in remote sensing Earth observation. Future work will explore the fields of video object segmentation and scene recognition.

CRediT authorship contribution statement


Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Sincerely thanks to the editors and reviewers for their valuable comments and suggestions. This work was supported by the National Natural Science Foundation of China under Grant 42230108 and 42271411. We are grateful to the institutions and companies for providing the satellite data.

References


