Received 27 November 2022, accepted 16 December 2022, date of publication 19 December 2022, date of current version 23 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3230826

RESEARCH ARTICLE

Efficient Real-Time Tracking of Satellite Components Based on Frame Matching

HAO ZHANG[1], YANG ZHANG[1], JINGMIN GAO[1], HONGBO YANG[1], AND KEBEI ZHANG[2]
[1]School of Automation, Beijing Information Science and Technology University, Beijing 100192, China
[2]Beijing Institute of Control Engineering, Beijing 100094, China

Corresponding author: Yang Zhang (zhangyang@bistu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 62001035, in part by the Research and Development Program of Beijing Municipal Education Commission under Grant KM202111232018, and in part by the Key Research Training Project of Beijing Information Science and Technology University under Grant 2021YJPY216.

ABSTRACT In order to obtain the satellite's in-orbit attitude information, it is necessary to track the satellite components in satellite video sequences. To solve the problem of low illumination and target occlusion in space environment, we propose an efficient satellite component tracking technique based on Rethinking Space-Time Networks with Improved Memory Coverage (STCN). We classify the pixels in the query frame by feature matching network that establishes the corresponding relationship between the frames. Unlike STCN, we reduce the contribution of background region in feature matching and enhance the robustness of the model in low illumination environment, thus improving the segmentation results. For lost targets due to the overturning and occlusion of satellite components, a position information encoder module is designed to further raise the tracking performance of the model. In addition, we present a local matching module to upgrade the existing feature matching methods. Experiments demonstrate that compared to STCN, our method heightens the tracking performance (J&F) by 10.1% and can achieve multi-object recognition at 15+ FPS.

INDEX TERMS Tracking, video object segmentation, deep learning, low-light, target occlusion.

I. INTRODUCTION

With the rapid development of spacecraft technology, people pay more and more attention to the tasks of target satellite identification, tracking and attitude estimation. It has become an important development trend in the field of satellite technology in various countries to vigorously develop information acquisition and processing technology related to satellites and other aircraft [1], [2], [3]. Among them, target detection and recognition, whose main content is to accurately identify the types of space targets and effectively invert the target attributes such as satellite geometry size, is an important prerequisite and guarantee technology for satellite docking.

In recent years, vision-based satellite component tracking and detection technology have attracted much attention because of its advantages of simple implementation and low power consumption. Especially in practical applications, video sequences are the mainstay. There are three main problems in tracking and detecting satellite components:

(1) When the satellite is in orbit, the illumination intensity of the satellite is constantly changing due to its constantly changing orbital position. Especially when the satellite runs to the back of the earth, its illumination intensity is low, and the satellite local components in the video sequence will be difficult to observe.

(2) In the process of tracking and detecting satellite local components, the satellite may turn over and occlusion, resulting in the loss of the tracked target and the decline of detection accuracy.

(3) When observing the target satellite, because the target satellite is always in motion, it may cause the image captured by the imaging equipment to shake and blur, resulting

The associate editor coordinating the review of this manuscript and approving it for publication was Rosario Pecora.

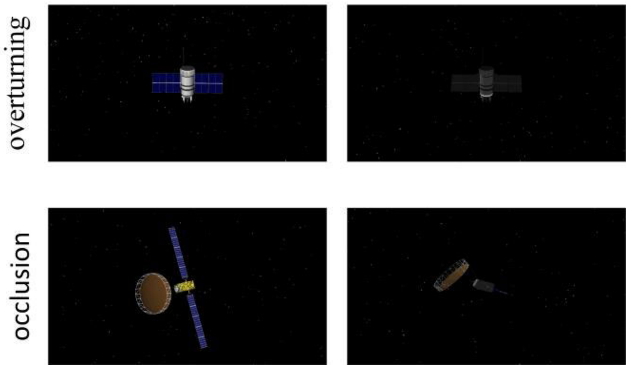

FIGURE 1. Schematic diagram of satellite rollover and occlusion.

in unclear imaging and interference with the tracking and detection of satellite local components.

The above problems pose great challenges to the tracking and detection of satellite local components.

In this paper, the satellite tracking task is realized by video object segmentation (VOS) [6], [7], [8] of satellite components. VOS belongs to the field of computer vision and has important applications in many fields. In this work, we focus on the VOS of semi-supervised satellite components, in which the ground truth segmentation masks of one or more objects are given for the first frame in the video. With the rapid development of deep learning and the introduction of DAVIS data-set [4], [5], The task of semi-supervising VOS has also made great progress in recent years. Many early studies used online learning strategies to fine-tune the corresponding network by giving the first frame mask [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20]. This method has high accuracy, but it takes a long time to infer.

On the basis of ensuring accuracy, recent works are faster than the online fine-tuning method. Space-time memory networks (STM) [26] introduces a memory network for the first time, improves the accuracy of video segmentation by storing the features of historical frames, and introduces the concept of global matching. Collaborative video object segmentation by foreground-background integration (CFBI) [23] heightens the segmentation effect by comparing the features of foreground and background and has a good effect on multi-scale targets. Fast end-to-end embedding learning for video object segmentation (FEELVOS) [24] uses global matching and local matching mechanisms for each frame to segment the current frame through information transmission. Kernelized memory network (KMN) [25] uses static images to pre-train the model, and Gaussian kernel is introduced to enhance the effectiveness of the memory network. Memory-augmented self-supervised tracker (MAST) [27] proposes a self-supervised training model, which achieves the same performance as the supervised methods without any annotations. Mask Selection Network (MSN) [28] uses temporal consistency to forward and reverse the video sequences, and uses the difference of masks given by them to correct the network, which effectively suppresses noise and achieves extremely high accuracy. Reliable Propagation-Correction Modulation for Video Object Segmentation (RPCMVOS) [29] corrects the noise propagating in the network from the local correlation frame and the global correlation frame by introducing the propagation modulator and the correction modulator. Multimodal Transformers (MTTR) [30] uses the transformer to model the VOS task as a sequence prediction problem, which greatly simplifies the model and has impressive results in multiple metrics. The above methods have achieved impressive results in accuracy, but it is difficult to meet the real-time requirement of satellite component tracking. Learning fast and robust target models (FRTM) [31] adopts a new network structure composed of two lightweight modules, which combines online learning and offline learning at the same time, achieving high frame rate and good performance. Hierarchical Memory Matching Network (HMMN) [32] proposed a new memory module, which effectively utilized the time smoothness, classified the memory, and realized more accurate memory matching. Rethinking Space-Time Networks with Improved Memory Coverage (STCN) [21] forms an efficient and robust framework by establishing the corresponding relationship between frames. However, the robustness of this model to the rollover and occlusion of satellite construction is poor. Pixel-Level Bijective Matching for Video Object Segmentation (BMVOS) [22] introduces a bijective matching mechanism to make every pixel have a chance to contribute. Although this method has a fast-processing speed, it is difficult to meet the aerospace requirements in terms of accuracy.

In this work, we take STCN [21] network with the best accuracy and speed as our backbone network. STCN is a simple, effective and efficient framework for video object segmentation. We propose an attention module with shared weight to improve the detection ability of the model for satellite components in low illumination environment. Through this attention module, we can extract the features of satellite components in the image more effectively, and distinguish the foreground from the background. To enhance the tracking accuracy of the model for the overturning and occlusion of satellite components, we introduce a position information coding scheme and propose a local matching module based on transformer [33]. By capturing the position information of satellite components in video images, it can effectively avoid the problem of target loss caused by the change of gray scale, shape and other features caused by the overturning and occlusion of satellite components. Because the changes of adjacent frames in the video sequence are small, our local matching module can match features from adjacent frames, thus bringing more excellent performance and more efficient memory usage.

The contributions of this paper are summarized as follows:

1. We propose a video object segmentation model for satellite component tracking, which is more robust to low illumination in space environment, overturning and blocking of satellite components than STCN.

2. We propose three simple and efficient modules: attention sharing module, position information encoder and local matching module.
3. Our network can reach the speed of 15+ FPS while maintaining high efficiency.

II. SATELLITE COMPONENT TRACKING METHOD

Given the satellite component mask of the first frame of a video sequence, we process the satellite component masks of subsequent frames in sequence. We store the features of keyframes through memory storage, compare the features of the current frame with those of keyframes, and then classify each pixel of the current frame.

A. OVERALL FRAMEWORK

The overview of our framework is illustrated in Figure 2. We adopted STCN [21] as the baseline backbone. As with STCN, we use resnet50 [34] as key encoder to get image features, and resnet18 as value encoder to get mask features. Resnet, as a classic feature extraction network, is more robust to few-shot learning than the newly proposed Swin-Transformer [49] and EfficientNet [50]. In addition, image features are more complex, and difficult to extract than mask features, so a deeper network is needed to extract image features.

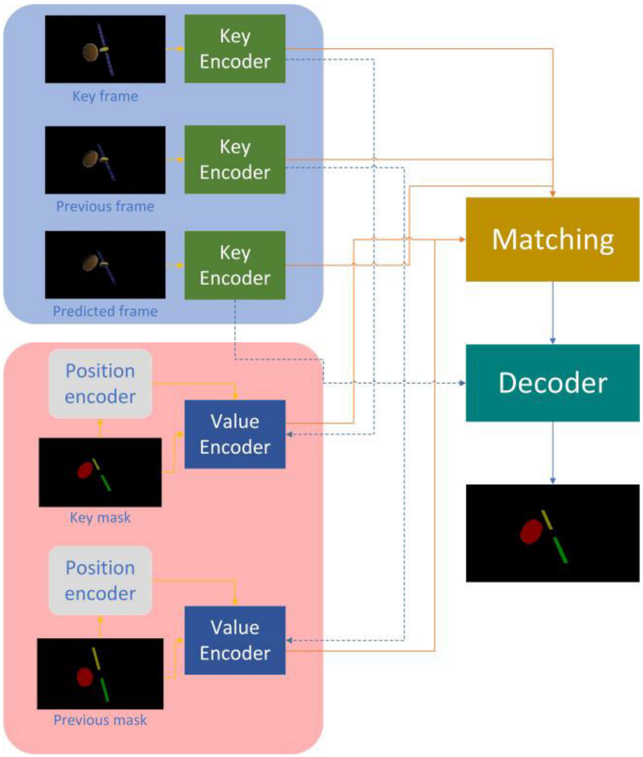

FIGURE 2. Network structure of satellite component tracking.

We add the attention module to the key encoder, which can help us clearly distinguish the foreground from the background in the image, and make the model connect with the features of satellite components in the subsequent feature matching. We add a position information encoder to the value encoder. This module allows us to get the position of the corresponding satellite component in this frame. By sending the corresponding mask and the position information into the value encoder, a more accurate mask feature map can be obtained.

Figure 3 shows the inferring process of our method. The memory in the figure contains the feature map of the key frame and the previous frame. The position information is embedded into the feature map of the memory bank through the position information module. The current frame gets the corresponding feature map through encoder. After that, the feature map obtained from the current frame is sent to the attention sharing module for local and global matching with the feature map in the memory. Finally, the matched result is sent to a decoder to obtain a mask.

We send the feature maps of the keyframe and the previous frame into the global matching module and the local matching module, respectively. By comparing the features of the current frame and keyframe, the feature query of the current frame and previous frame searches all possible key feature combinations. The final matching score is obtained by linearly adding the global matching result and the local matching result. We will introduce the local matching module later.

The global feature matching formula is as follows:

$$S_{ij}^{L2} = 2k_i^M \cdot k_j^Q - \left\| k_i^Q \right\|_2^2 \tag{1}$$

where $\boldsymbol{M}$ represents the keyframe in memory, which we collectively call the memory frame, $\boldsymbol{Q}$ represents the current frame, $\boldsymbol{K}$ represents the feature matrix obtained by key encoder, $\boldsymbol{i}$ and $\boldsymbol{j}$ respectively represent the positions of the memory frame and the current frame in the video sequence, $\boldsymbol{S}$ represents the correlation degree, and $\boldsymbol{L2}$ represents the Euclidean distance. Finally, the matching matrix of the memory frame and the current frame can be obtained, and the result can be obtained by matrix multiplication calculation of the matching matrix and the mask features in the memory module:

$$V^I = V^M S_{ij}^{L2} \tag{2}$$

where $\boldsymbol{V^M}$ represents the mask feature matrix in the memory module. Finally, the memory characteristic matrix $\boldsymbol{V^I}$ is passed to the decoder to generate a mask.

B. ATTENTION SHARING MODULE

At present, attention mechanism [33] is widely used in various fields of deep learning. Attention mechanism is used in most unsupervised VOS tasks [35], [36], [37], [38], [39], but rarely used in semi-supervised tasks. The images captured during the satellite docking process have the characteristics of single background and low illumination. By adding attention module, the foreground and background in the video can be effectively distinguished, and the robustness of the model to low illumination environment can be enhanced.

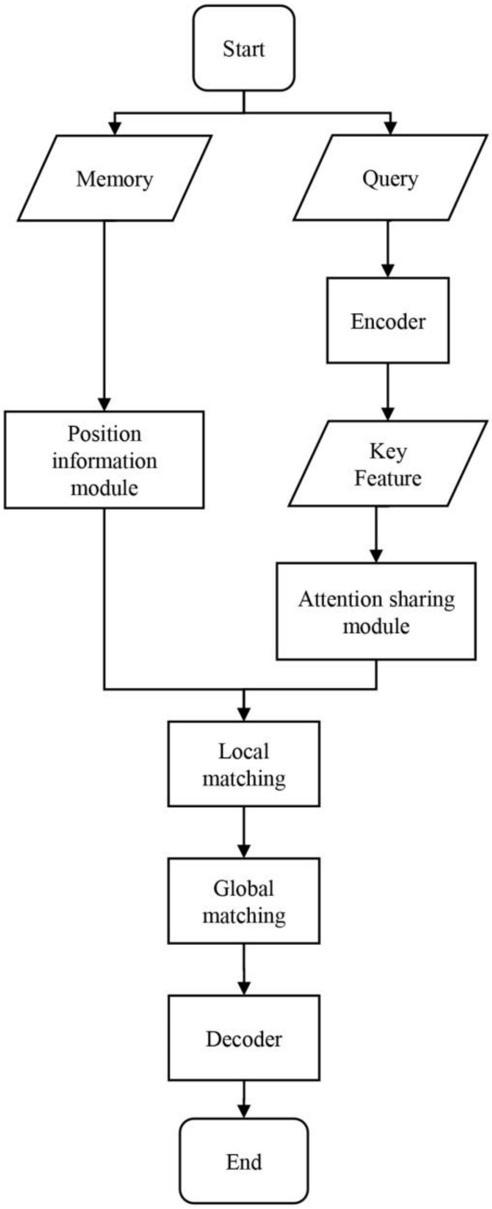

FIGURE 3. Process of satellite components tracking.

Our attention sharing module is shown in Figure 4. The feature matrix obtained by the key encoder takes the maximum, average and variance along the channel dimension, and a new feature matrix is obtained. The feature matrix passes through a 7 × 7 convolution kernel to obtain a weight matrix with channel 1, which represents the attention distribution probability of the frame. Finally, the weight matrix is used to point multiply the key features and the corresponding mask features respectively.

As shown in Figure 5, the abscissa in the figure represents the number of pixels, and the ordinate represents the variance of the corresponding pixels along the channel direction. The red dotted line represents the pixels of satellite components, and the blue solid line represents the background pixels. It can be clearly seen from the figure that the characteristic variance of satellite components is generally larger than that of the background.

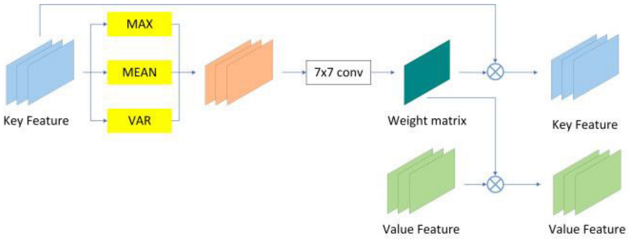

FIGURE 4. Attention sharing module.

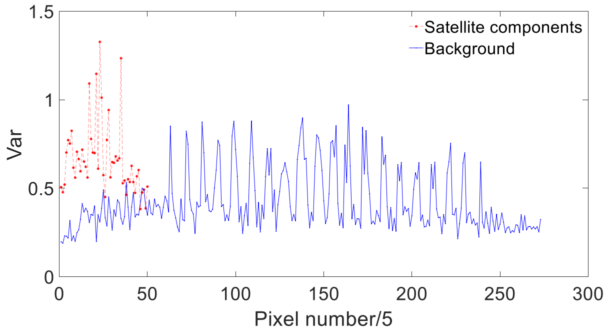

FIGURE 5. The variance of the channel direction of the corresponding pixel point of the feature map.

Figure 6 shows our attention map. The left side of the figure shows the satellite image, and the right side shows the corresponding attention heat map. We show a total of three groups of probability maps of attention distribution of images. We use the variance of the channel dimension of the feature map as one of the output features. We consider that the background of the satellite image is single and the variance of the corresponding features is small, while the features of the satellite components are complex and the variance of the corresponding features is large. Using variance value as output feature can effectively improve the ability of the model to distinguish foreground from background.

C. POSITION INFORMATION ENCODER

For the tracking task of satellite components, satellite components often turn over and occlusion. If the appearance information is used in feature matching, it is easy to be visually disturbed, because it is completely based on the visual information. To alleviate this problem, we propose a position information encoder. In fact, the deep convolution neural network itself has a certain ability to encode absolute position information [40], [41], [42], but it is very limited. In the field of machine translation, the relative position information in the sequence is effectively encoded by the extended self-attention mechanism [37], and in the field of object detection some people introduce the position information into transformer [44]. However, there is no position information coding method suitable for satellite component tracking.

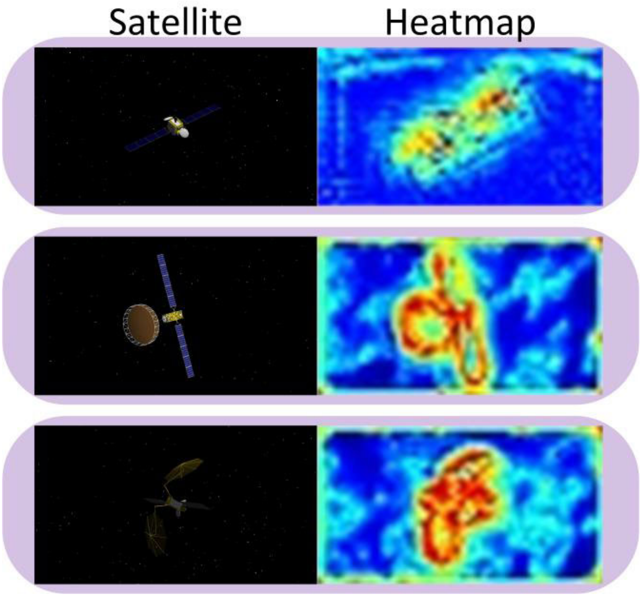

FIGURE 6. Probability map of attention distribution of satellite images.

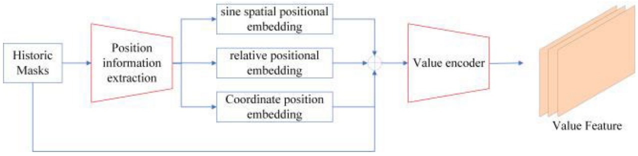
Historic Masks
Position information extraction
sine spatial positional embedding
relative positional embedding
Coordinate position embedding
Value encoder
Value Feature

FIGURE 7. Architecture of position information encoder.

Figure 7 shows the architecture of the position information encoder. As shown in the figure, after the historic mask passes through the position information extraction module, there are three kinds of position information encoding results: sine spatial positional embedding, relative positional embedding and coordinate position embedding. The sine spatial positional embedding refers to End-to-end object detection with transformers (DETR) [44], which is composed of sinusoidal functions with different frequencies. The difference is that we don't embed the position code by linear addition. To retain the original mask features fully, we embed the position code by splicing the mask and location information. Other position information is embedded in the same way. Relative position embedding is a learnable position information matrix with 64 channels, which is transformed into the same size as the mask feature map by linear interpolation. We use three coordinate matrices to form coordinate position embedding, which respectively represent the position change on the X axis, the position change on the Y axis and the position change of the polar coordinate with the center as the origin. Our position information embedding module is defined as:

$$P = P^S \oplus P^R \oplus P^C \tag{3}$$

where $\oplus$ indicates catting in the channel dimension. By embedding the above three position information matrices, the position information of the satellite components in the current frame can be effectively obtained, and a more accurate feature matrix can be provided for the subsequent feature matching links.

D. LOCAL MATCHING MODULE

Global matching is responsible for comparing the feature information of keyframes in a video sequence with the current frame, while local matching is responsible for comparing the feature information of the current frame position in a video sequence in spatial-temporal neighborhood. Global matching has the advantages of simple implementation and higher reasoning speed. Moreover, in feature matching, because there are many keyframes, global matching has strong robustness to the wrong feature matching in partial keyframes. However, there is no concept of time consistency in global matching. If the appearance information of the segmentation target is similar to that of the background, or the appearance information of different segmentation targets is similar, it will probably lead to wrong segmentation results. This is fatal to the tracking of satellite components. Because in the process of satellite component segmentation, segmentation objects with very similar appearance information often appear, such as two very similar solar panel wings or antennas. Local matching mainly focuses on the information in the spatial-temporal neighborhood of each current frame position. Because the image changes little in the adjacent video frames, especially the position information changes. Therefore, it is more efficient to deal with local matching of similar targets.

In order to enhance the detection accuracy of the model for targets with similar appearance, we not only added the position information encoding module, but also proposed the local matching module. Several existing works also use local matching [23], [24], [43] or optical flow [45] to improve the segmentation accuracy of the model. However, few people use transformer to design local matching module. Our local matching module is defined as follows:

$$k_Q = MLP(f_Q) \tag{4}$$
$$k_M = MLP(f_M) \tag{5}$$
$$v = MLP(v_M) \tag{6}$$
$$S_{\text{loc}} = k_Q \cdot k_M^{p(i)} + P(k_Q) \tag{7}$$
$$loc = S_{loc} \cdot (v + v_p) \tag{8}$$

where $\boldsymbol{f_Q}$ represents the feature map of current frame, $\boldsymbol{f_M}$ represents the feature map of memory frame, $\boldsymbol{v_M}$ represents the memory mask feature map, $\boldsymbol{S_{loc}}$ represents the local affinity matrix, $\boldsymbol{k_Q}$ and $\boldsymbol{k_M}$ represent the feature vectors of the current frame and the memory frame, respectively. The current frame $\boldsymbol{f_Q}$ and the memory frame $\boldsymbol{f_M}$ get the feature matrix through the key encoder, and the feature matrix gets $\boldsymbol{k_Q}$ and $\boldsymbol{k_M}$ through two full connection layers respectively. Like transformer, $\boldsymbol{k_Q}$ and $\boldsymbol{k_M}$ are query values and key values respectively. $\boldsymbol{p(i)}$ represents the spatial neighborhood with the pixel $\boldsymbol{i}$ of the corresponding query frame as the center in the memory frame. $\boldsymbol{P(k_Q)}$ and $\boldsymbol{v_p}$ are the relative position embedded information [37] in the local affinity matrix and the

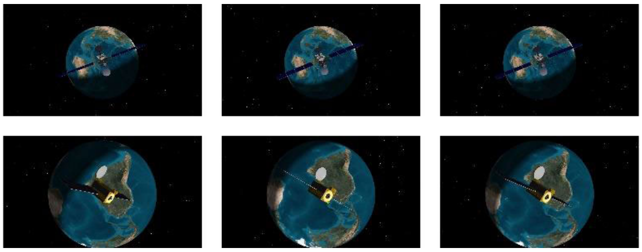

FIGURE 8. A random subset of video sequence satellite components dataset.

memory mask feature map, respectively. The final matching result ***loc*** is achieved by dot multiplication of the local affinity matrix and the mask feature map.

III. EXPERIMENTS

In this section, we describe the experimental results obtained from this study. In section A, we introduced our datasets and evaluation metrics. In section B, we introduced our training details. Section C shows the performance of our model in different scenarios, and compares our model with the latest methods. In order to verify the effectiveness of each module proposed by us, we conducted extensive ablation experiments in Section D. In section E, we give the detailed running time of each component of the model.

A. DATASETS AND EVALUATION METRICS

The dataset of satellite images we use comes from Systems Tool Kit (STK), which is the world's top satellite simulation software produced by AGI Company of the United States. The version of STK we use is 10. We mainly use STK to provide a high-precision visual simulation module, which can provide users with high-fidelity visual support in space. We collected 18 video sequences as our training set and 6 video sequences as our verification set. A video sequence contains a satellite, and each video sequence contains 40 to 60 satellite images.

Figure 8 shows a partial dataset image. In order to ensure the validity of our data set, while simulating the satellite docking scene, there are many images in our data set when the satellite turns over and blocks. We choose the solar wing and antenna as our tracking objects. First, these two satellite components exist in almost all satellites, so to estimate the attitude of the satellites during docking, has high universality. Secondly, the solar wing and antenna always exist in pairs. However, for many of the most advanced models at present, it is difficult to accurately distinguish two similar tracking objects, and the tracking of similar objects is very common in the tracking of satellite components.

We use $\boldsymbol{J\&F}$, an evaluation index commonly used in VOS. $\boldsymbol{J}$ score is calculated as the average Intersect over Union (IoU) score of prediction mask and the ground truth mask, which describes the accuracy in the whole mask area. $\boldsymbol{F}$ score is calculated as the average boundary similarity measure between the prediction mask and the ground truth mask, which describes the accuracy of the object boundary. $\boldsymbol{J\&F}$ refers to calculating the average value between them. The evaluation metric is defined as follows:

TABLE 1. Satellite component dataset. We marked thousands of pictures for testing and training. All our video sequences contain the situation that the satellite components turn over. The overturning of satellite components in the table is defined as: in a video sequence, the color information of the tracked object changes, and the occlusion of satellite components is defined as: in a video sequence, there are some frames where the tracked object does not appear.

Number	Training set	Test set
All video sequences	18	6
All video images	948	345
Video sequences containing satellite component turn over	18	6
Video sequences containing the occlusion of satellite components	8	3

$$J = \frac{S_p \cap S_{GT}}{S_p \cup S_{GT}} \tag{9}$$

$$F = \frac{2B_p \cdot B_r}{B_p + B_r} \tag{10}$$

$$J\&F = \frac{J + F}{2} \tag{11}$$

where $\boldsymbol{S_p}$ represents the prediction result of the model and $\boldsymbol{S_{GT}}$ represents the ground truth. $\boldsymbol{B_p}$ and $\boldsymbol{B_r}$ respectively represent the precision and recall of the prediction mask relative to the ground truth. $\boldsymbol{B_p}$ and $\boldsymbol{B_r}$ are defined as follows:

$$B_p = \frac{P_T}{P_{all}} \tag{12}$$

$$B_r = \frac{P_T}{P_{GT}} \tag{13}$$

where $\boldsymbol{P_T}$ is the number of boundary elements correctly predicted by the model, $\boldsymbol{P_{all}}$ is the total number of boundary elements predicted by the model, and $\boldsymbol{P_{GT}}$ is the total number of boundary elements with the ground truth.

B. TRAINING DETAILS

We use an 11GB 2080Ti GPU with the Adam optimizer [46] using PyTorch [47] to train our model. In the process of data preprocessing, firstly, we reduce the short edge of the image to 480 pixels, which can effectively speed up the training and inferring of the model with little impact on the accuracy. After that, we will randomly flip the image horizontally and shake the color. We use a batch size of 4 and 3000 iterations during training. In each iteration, we select three time-sequential frames from a video sequence, with the first frame as the starting frame to form a globally matched training sample. Then, the previous frame of the last two frames is taken as a training sample for local matching. A total of five images are taken from the video sequence. First, we use the globally matched first frame and the corresponding locally matched frame to predict the second frame, and then use the first and second frames as global matches to predict the third frame together with the locally matched frame of the third frame.

The momentum of our Adam optimizer is set to $\beta_1 = 0.9$, $\beta_2 = 0.999$, the basic learning rate is 10^{-5}, and the L2 weight decay of 10^{-7}. Our learning rate decays with a decay ratio of $\gamma = 0.1$. We use cross entropy as our loss function. In addition, in training, we choose the top-$p\%$ pixels with the highest loss to carry out back propagation. p is 100 in the first 1000 iterations, then linearly decreases to 15 in 1000 to 2000 iterations, and finally remains unchanged. Our model doesn't need to set any hyperparameters when inferring.

Figure 9 shows the change of loss function of three methods in training. It can be seen that the learning efficiency of the three methods is excellent. However, after 1000 iterations, the loss of our model is obviously lower than the other two. This shows that our model has better performance for satellite component tracking.

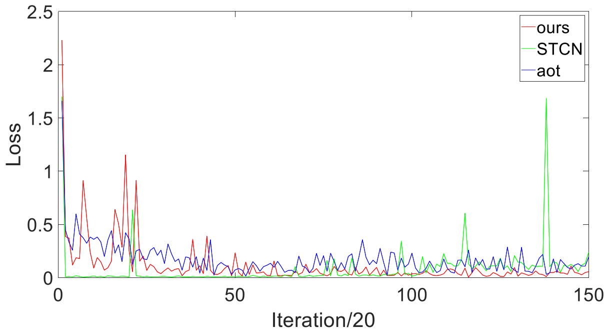

FIGURE 9. Total loss curve in different model during training.

Because it is difficult for us to obtain a large number of labeled satellite images, we use transfer learning to solve the problem of few-shots learning. Specifically, we use STCN [21] pre-trained parameters to initialize the network skeleton. Standard normal distribution initialization parameters are used for attention module, position information module and local matching module. Moreover, Siamese network [48] is used in the backbone structure of our network, which is more robust to few-shots learning.

Figure 10 shows the influence of the pre-training model on the loss during training. It can be seen from the figure that the model is easier to converge after using the pre-training model. In addition, the basic learning rate of the model with no pre-training is 10^{-6}, and the other hyperparameters are the same as the model with pre-training. We choose different basic learning rates and optimizers to train our models with no pre-training. Most of these models can hardly converge, and even the gradient explosion with loss value of **NaN** will occur. Using the pre-training model can not only make the training easy, but also improve the performance of the model [51]. Especially for few-shot learning, it is very important to use pre-training model.

We use data augmentation to enrich our dataset. Specifically, we perform the same color jitter of (brightness=0.1, contrast=0.03, saturation=0.03), and random gray scale with a probability of 0.05 on the extracted images in the video sequence. After that, we perform color jitter of (brightness=0.01, contrast=0.01, saturation=0.01), and perform different random affine on each image.

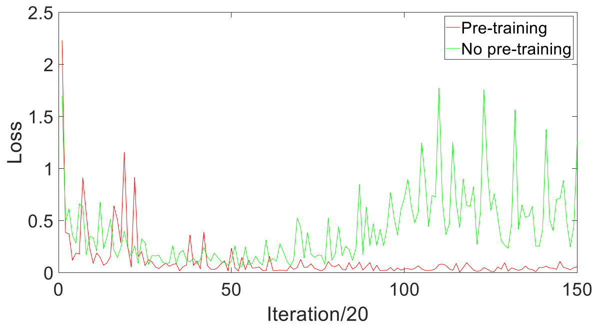

FIGURE 10. Comparison of loss curves between pre-training and no pre-training.

C. EVALUATIONS

Table 2 tabulates the comparison between our method and the most advanced methods in VOS segmentation benchmark at present. We made a comparison on three standard metrics: region similarity $\boldsymbol{J}$, contour accuracy $\boldsymbol{F}$ and average score $\boldsymbol{J}\&\boldsymbol{F}$. For the calculation of model speed, we calculate multi-object FPS, which is defined as the total number of output masks divided by the total time for the model to process all images. For the speed of comparison methods, we use the same device to measure according to the above standards.

TABLE 2. Comparisons between different methods. All models are trained on the same device using our dataset, and the corresponding pre-training weights are loaded before training. All three methods are iterated 3000 times.

Method	fps	$\boldsymbol{J}$	$\boldsymbol{F}$	$\boldsymbol{J}\&\boldsymbol{F}$
STCN	20.2	72.4	84.7	78.5
aot	12.1	76.5	84.8	80.6
Ours	15.1	83.5	93.7	88.6

TABLE 3. Performance of models in different scenarios.

	$\boldsymbol{J}$	$\boldsymbol{F}$	$\boldsymbol{J}\&\boldsymbol{F}$
Rollover	72.9	97.5	85.2
Occlusion	80.1	89.4	80.6
Background is the earth.	90.2	96.4	93.3
Low illumination	90.8	91.4	91.1

As shown in Table 3, we listed the performance of our model in different scenarios. It is obvious from the table that our model performs well in common satellite scenes. Our model is not only robust to the turning and blocking of satellites, but also shows excellent performance in low illumination and complex background.

Figure 11 shows the comparison between our model and other models in terms of speed and accuracy. Although our model is inferior to STCN in speed, our model is far superior to other models in evaluation metric $\boldsymbol{J}\&\boldsymbol{F}$. Figure 12 shows

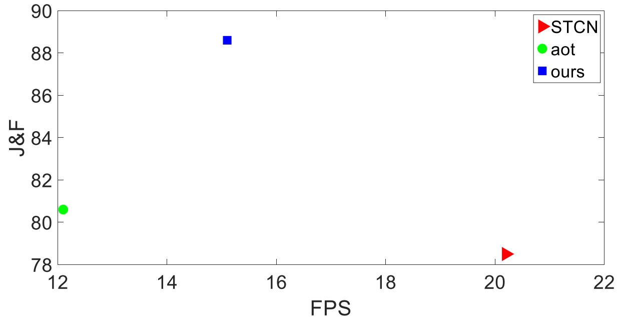

FIGURE 11. Comparison of speed and accuracy of different methods.

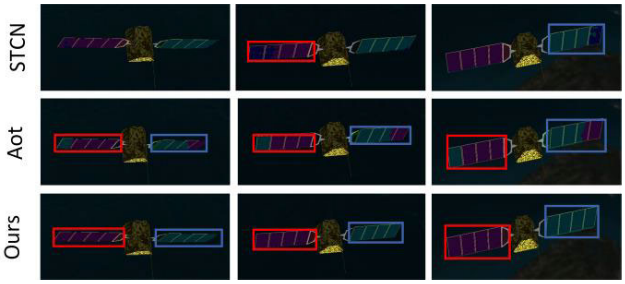

FIGURE 12. Visualization of semi-supervised VOS results with the first column being the reference masks to be propagated.

how our model compares with other models during mask propagation. The red box and blue box in the figure indicate the effect comparison of the left wing and the right wing of the satellite under different models. It can be seen that other models are difficult to accurately distinguish satellite components with similar appearances since they have no specific design for processing low illumination and highly similar objects. Figure 13 shows the comparison of the effects of different models in the case of satellite component overturning and occlusion. GT in the figure represents the ground truth. Due to its attention sharing module, position information encoder and local matching module, our model is more effective for the overturning and occlusion of satellite components compared to other models.

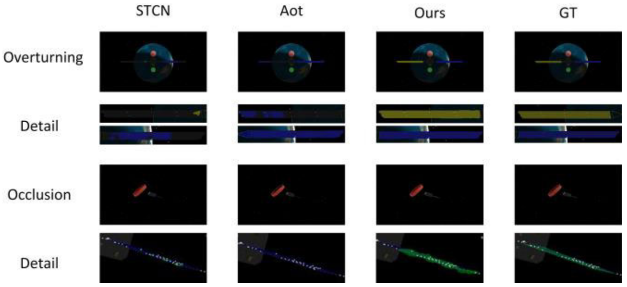

FIGURE 13. Visualization of satellite tracking results under rollover and occlusion.

D. ABLATION STUDY

In Table 4, we analyze the effect of each module on the model. The comparison of speed and accuracy between different modules can be seen more intuitively from Figure 14. As can be seen, enabling a certain module alone has a very limited improvement in accuracy. Among them, we are most interested in the local matching module. One of the reasons is that this module occupies the most computing resources. In the case of taking up a lot of computing resources, the improvement of model accuracy by local modules is very limited, but if it is enabled with other modules at the same time, the performance of the model can be greatly enhanced. In short, location information and attention information play a greater role in local matching. We speculate that this is because the gap between memory frame and query frame is small in local matching, and the performance of similar location information and similar attention weight matrix is stronger in matching.

TABLE 4. Ablation study of three modules. ATT denotes attention sharing module, POS denotes location information module, and LOC denotes local matching module.

	ATT	POS	LOC	fps	J	F	J&F
1	√	√	√	15.1	83.5	93.7	88.6
2	√			19.6	77.5	83.7	80.6
3			√	16.2	78.5	84.6	81.5
4	√		√	16.1	84.0	89.8	86.9
5		√	√	15.9	83.8	89.1	86.4

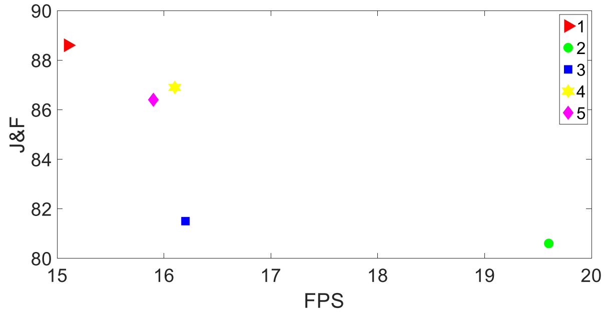

FIGURE 14. Visualization of speed and accuracy between different modules.

As shown in Table 5, the performance of the model with pre-training is far superior to that without pre-training. This is because the model can hardly converge without loading pre-training weights. It is difficult to learn some basic texture and color features only by our data set, especially for encoders that need to extract features.

TABLE 5. Influence of using pre-training model on results.

	J	*F*	*J&F*
Pre-training	83.5	93.7	88.6
No pre-training	36.5	54.0	45.2

E. ANALYSIS OF RUNNING TIME AND REAL-TIME PERFORMANCE

We analyze the running time of each component of the two models in Figure 15. The running time in the graph repre-

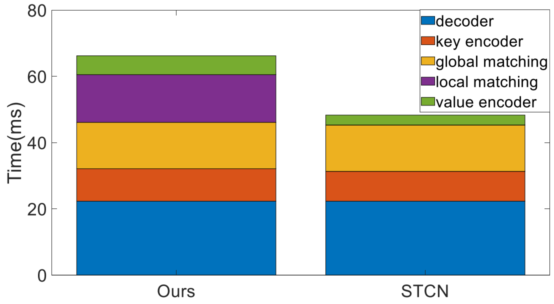

FIGURE 15. **The average running time of our model and STCN in each component respectively.**

sents the time required for the model to process one image. Although the speed of our method is lower than that of STCN, it can still reach the processing speed of 66.2 milliseconds. Our method can be used in real time. We add a local matching module which takes a lot of computing resources to improve the performance of the model, as described in the ablation experiment.

IV. CONCLUSION

We propose a simple and efficient tracking framework for satellite components. We propose the attention sharing module, which can effectively improve the performance of the model and solve the problem of low illumination tracking in space environment. Our proposed location information module and local matching module effectively solve the problem of tracking target loss caused by the overturning and occlusion of satellite components. Compared with the most advanced methods at present, our method has more excellent performance in tracking satellite components. However, when the tracking target in the video sequence is lost for a long time and the components of the satellite are overturned, the local matching module will fail because there is no corresponding tracking target in the previous frame, and the ideal performance may not be achieved only by global matching. Moreover, in the docking task of satellites, higher speed and lighter weight models are needed. We will further complete it in the future work.

REFERENCES

[1] Y. Chen, J. Gao, and K. Zhang, "R-CNN-based satellite components detection in optical images," *Int. J. Aerosp. Eng.*, vol. 2020, pp. 1–10, Oct. 2020.

[2] J. Wang, W. Liu, H. Wang, L. Zukun, and O. Gang, "An attitude estimation algorithm for satellite navigation array against gross error," in *Proc. China Satell. Navigat. Conf. (CSNC)*. Cham, Switzerland: Springer, 2021, pp. 473–482.

[3] B. Chen, J. Cao, A. Parra, and T.-J. Chin, "Satellite pose estimation with deep landmark regression and nonlinear pose refinement," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 1–9.

[4] S. Caelles, J. Pont-Tuset, F. Perazzi, A. Montes, K.-K. Maninis, and L. Van Gool, "The 2019 DAVIS challenge on VOS: Unsupervised multi-object segmentation," 2019, *arXiv:1905.00737*.

[5] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool, "The 2017 Davis challenge on video object segmentation," 2017, *arXiv:1704.00675*.

[6] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 724–732.

[7] S. Caelles, A. Montes, K.-K. Maninis, Y. Chen, L. Van Gool, F. Perazzi, and J. Pont-Tuset, "The 2018 DAVIS challenge on video object segmentation," 2018, *arXiv:1803.00557*.

[8] S. Sukhbaatar, J. Weston, and R. Fergus, "End-to-end memory networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1–9.

[9] J. S. Yoon, F. Rameau, J. Kim, S. Lee, S. Shin, and I. S. Kweon, "Pixel-level matching for video object segmentation using convolutional neural networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2167–2176.

[10] J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang, "SegFlow: Joint learning for video object segmentation and optical flow," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 686–695.

[11] F. Perazzi, A. Khoreva, R. Benenson, B. Schiele, and A. Sorkine-Hornung, "Learning video object segmentation from static images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2663–2672.

[12] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixe, D. Cremers, and L. Van Gool, "One-shot video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 221–230.

[13] Y.-T. Hu, J.-B. Huang, and A. Schwing, "MaskRNN: Instance level video object segmentation," in *Proc. Adv. Neural inf. Process. Syst.*, vol. 30, 2017, pp. 1–10.

[14] H. Ci, C. Wang, and Y. Wang, "Video object segmentation by learning location-sensitive embeddings," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 501–516.

[15] L. Bao, B. Wu, and W. Liu, "CNN in MRF: Video object segmentation via inference in a CNN-based higher-order spatio-temporal MRF," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5977–5986.

[16] J. Luiten, P. Voigtlaender, and B. Leibe, "PReMVOS: Proposal-generation, refinement and merging for video object segmentation," in *Proc. Asian Conf. Comput. Vis.* Cham, Switzerland: Springer, 2018, pp. 565–580.

[17] K.-K. Maninis, S. Caelles, Y. Chen, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool, "Video object segmentation without temporal information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 6, pp. 1515–1530, Jun. 2019.

[18] P. Voigtlaender and B. Leibe, "Online adaptation of convolutional neural networks for video object segmentation," 2017, *arXiv:1706.09364*.

[19] Z. Wang, J. Xu, L. Liu, F. Zhu, and L. Shao, "RANet: Ranking attention network for fast video object segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3978–3987.

[20] X. Li and C. C. Loy, "Video object segmentation with joint reidentification and attention-aware mask propagation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 90–105.

[21] H. K. Cheng, Y.-W. Tai, and C.-K. Tang, "Rethinking space-time networks with improved memory coverage for efficient video object segmentation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 11781–11794.

[22] S. Cho, H. Lee, M. Kim, S. Jang, and S. Lee, "Pixel-level bijective matching for video object segmentation," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 129–138.

[23] Z. Yang, Y. Wei, and Y. Yang, "Collaborative video object segmentation by foreground-background integration," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 332–348.

[24] P. Voigtlaender, Y. Chai, F. Schroff, H. Adam, B. Leibe, and L.-C. Chen, "FEELVOS: Fast end-to-end embedding learning for video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9481–9490.

[25] H. Seong, J. Hyun, and E. Kim, "Kernelized memory network for video object segmentation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 629–645.

[26] S. W. Oh, J.-Y. Lee, N. Xu, and S. J. Kim, "Video object segmentation using space-time memory networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9226–9235.

[27] Z. Lai, E. Lu, and W. Xie, "MAST: A memory-augmented self-supervised tracker," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6479–6488.

[28] V. Goel, J. Li, S. Garg, H. Maheshwari, and H. Shi, "MSN: Efficient online mask selection network for video instance segmentation," 2021, *arXiv:2106.10452*.

[29] X. Xu, J. Wang, X. Li, and Y. Lu, "Reliable propagation-correction modulation for video object segmentation," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2022, vol. 36, no. 3, pp. 2946–2954.
[30] A. Botach, E. Zheltonozhskii, and C. Baskin, "End-to-end referring video object segmentation with multimodal transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 4985–4995.
[31] A. Robinson, F. J. Lawin, M. Danelljan, F. S. Khan, and M. Felsberg, "Learning fast and robust target models for video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7406–7415.
[32] H. Seong, S. W. Oh, J.-Y. Lee, S. Lee, S. Lee, and E. Kim, "Hierarchical memory matching network for video object segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 12889–12898.
[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.
[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[35] Z. Zhong, Z. Q. Lin, R. Bidart, X. Hu, I. B. Daya, Z. Li, W.-S. Zheng, J. Li, and A. Wong, "Squeeze-and-attention networks for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13065–13074.
[36] Y.-W. Chen, X. Jin, X. Shen, and M.-H. Yang, "Video salient object detection via contrastive features and attention modules," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 1320–1329.
[37] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," 2018, *arXiv:1803.02155*.
[38] W. Wang, H. Song, S. Zhao, J. Shen, S. Zhao, S. C. H. Hoi, and H. Ling, "Learning unsupervised video object segmentation through visual attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3064–3074.
[39] X. Lu, W. Wang, C. Ma, J. Shen, L. Shao, and F. Porikli, "See more, know more: Unsupervised video object segmentation with co-attention Siamese networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3623–3632.
[40] M. Amirul Islam, M. Kowal, S. Jia, K. G. Derpanis, and N. D. B. Bruce, "Position, padding and predictions: A deeper look at position information in CNNs," 2021, *arXiv:2101.12322*.
[41] M. A. Islam, M. Kowal, S. Jia, K. G. Derpanis, and N. D. B. Bruce, "Global pooling, more than meets the eye: Position information is encoded channel-wise in CNNs," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 793–801.
[42] M. Amirul Islam, S. Jia, and N. D. B. Bruce, "How much position information do convolutional neural networks encode?" 2020, *arXiv:2001.08248*.
[43] Y.-T. Hu, J.-B. Huang, and A. G. Schwing, "VideoMatch: Matching based video object segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 54–70.
[44] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 213–229.
[45] H. Xie, H. Yao, S. Zhou, S. Zhang, and W. Sun, "Efficient regional memory network for video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1286–1295.
[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[47] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, and A. Desmaison, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–12.
[48] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a 'Siamese' time delay neural network," in *Proc. Adv. Neural inf. Process. Syst.*, vol. 6, 1993, pp. 737–744.
[49] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10012–10022.
[50] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.
[51] L. Qiao, Y. Zhao, Z. Li, X. Qiu, J. Wu, and C. Zhang, "DeFRCN: Decoupled faster R-CNN for few-shot object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 8681–8690.

HAO ZHANG received the B.S. degree from Beijing Information Science and Technology University, Beijing, China, in 2020, where he is currently pursuing the M.S. degree with the College of Automation. His research interests include artificial intelligence, computer vision in aerospace, and tracking of satellite components.

YANG ZHANG received the B.S. and Ph.D. degrees from the Ocean University of China, Qingdao, China, in 2010 and 2016, respectively. From 2014 to 2016, he was a Visiting Scholar with the Department of Electrical and Computer Engineering, University of Miami, Coral Gables, FL, USA. From 2017 to 2019, he was a Postdoctoral Researcher at Tsinghua University, Beijing, China. He is currently an Associate Professor with the College of Automation, Beijing Information Science and Technology University, Beijing. His research interests include underwater vision, video coding, and segmentation of satellite components.

JINGMIN GAO received the M.S. and Ph.D. degrees from the Beijing Institute of Technology, Beijing, China, in 1995 and 1999, respectively. Since 1999, she has been a Professor with the College of Automation, Beijing Information Science and Technology University, Beijing. Her research interests include object detection and spacecraft pose estimation.

HONGBO YANG received the B.S. and M.S. degrees from the Hebei University of Technology, Hebei, China, in 1999 and 2002, respectively, and the Ph.D. degree from the Aerospace Information Research Institute, Chinese Academy of Sciences, China, in 2005. He is currently a Professor with the College of Automation, Beijing Information Science and Technology University, Beijing, China. His research interests include image processing, pattern recognition, and research on intelligent recognition of container images.

KEBEI ZHANG received the B.S. degree in automation from Beijing Information Science and Technology University, China, in 2014, and the Ph.D. degree from the Beijing Institute of Control Engineering, China, in 2018. His research interests include spacecraft attitude determination and attitude ultra-high accuracy attitude control.