Region Aware Video Object Segmentation With Deep Motion Modeling

Bo Miao, *Graduate Student Member, IEEE*, Mohammed Bennamoun, *Senior Member, IEEE*, Yongsheng Gao, *Senior Member, IEEE*, and Ajmal Mian, *Senior Member, IEEE*

***Abstract*— Current semi-supervised video object segmentation (VOS) methods often employ the entire features of one frame to predict object masks and update memory. This introduces significant redundant computations. To reduce redundancy, we introduce a Region Aware Video Object Segmentation (RAVOS) approach, which predicts regions of interest (ROIs) for efficient object segmentation and memory storage. RAVOS includes a *fast* object motion tracker to predict object ROIs in the next frame. For efficient segmentation, object features are extracted based on the ROIs, and an object decoder is designed for object-level segmentation. For efficient memory storage, we propose motion path memory to filter out redundant context by memorizing the features within the motion path of objects. In addition to RAVOS, we also propose a large-scale occluded VOS dataset, dubbed OVOS, to benchmark the performance of VOS models under occlusions. Evaluation on DAVIS and YouTube-VOS benchmarks and our new OVOS dataset show that our method achieves state-of-the-art performance with significantly faster inference time, *e.g.*, 86.1 $\mathcal{J}\&\mathcal{F}$ at 42 FPS on DAVIS and 84.4 $\mathcal{J}\&\mathcal{F}$ at 23 FPS on YouTube-VOS. Project page: ravos.netlify.app.**

***Index Terms*— Video object segmentation, multi-object dense tracking, feature matching.**

I. Introduction

VIDEO object segmentation (VOS) is a fundamental research topic in visual understanding, with the aim to segment target objects across entire videos. VOS enables machines to recognize the motion pattern, location, and boundaries of the target objects in videos [1], [2], [3], [4], which can foster a wide range of applications, *e.g.*, augmented reality, video editing, and robotics. This work focuses on semi-supervised VOS, where object masks given on the first-frame are used to segment and track objects in subsequent frames. A practical semi-supervised VOS method should be able to segment the objects of interest efficiently and accurately under challenging scenarios, such as occlusions, large deformations, similar appearances, and scale variations.

Recent semi-supervised VOS methods mainly follow one of two paradigms: detection-based [5], [6], [7] and memory-based [8], [9], [10], [11]. Detection-based methods often rely on online adaption to make the model object-specific, while memory-based methods adopt memory networks to memorize and propagate spatio-temporal features across frames for object segmentation. Methods in the latter paradigm have recently drawn significant research attention due to their exceptional accuracy. These methods either perform non-local matching [8], [11] or local-matching [12], [13] for mask propagation.

Although current memory-based methods have shown promising performance, memorizing and segmenting the entire features of one frame inevitably leads to redundant computations and slows down VOS. Some methods have attempted to accelerate VOS by using additional instance segmentation or detection networks [14], [15], template matching modules [16], [17], [18], or optical flow [19] to create regions of interest (ROIs) and then performing local segmentation. However, these local segmentation methods are either not accurate enough or still time-consuming given the additional computational overhead. Therefore, developing an effective method that avoids redundant computations and memory storage, while maintaining high segmentation accuracy is significant for improving the overall semi-supervised VOS performance.

In this paper, we propose a novel Region Aware Video Object Segmentation (RAVOS) approach, which enables multi-object tracking and ROI prediction to achieve fast and accurate semi-supervised VOS with less memory burden. First, a lightweight object motion tracker (OMT) is proposed to estimate the parameters of motion functions using the position information of instances in past frames for object tracking and ROI prediction, as shown in Fig. 1. To achieve efficient object segmentation, we extract object features based on the predicted ROIs and adopt a designed object decoder that employs object skip connections for object-level segmentation. Second, we propose motion path memory (MPM) to filter out redundant context by memorizing the features within the motion path of objects between consecutive frames. Hence, redundant segmentation and memory storage are alleviated significantly.

Manuscript received 19 July 2022; revised 14 September 2023; accepted 8 March 2024. Date of publication 29 March 2024; date of current version 5 April 2024. This work was supported by the Australian Research Council Industrial Transformation Research Hub under Grant IH180100002. The work of Ajmal Mian was supported by Australian Research Council Future Fellowship Award funded by Australian Government under Project FT210100268. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Adrian Munteanu. *(Corresponding author: Ajmal Mian.)*

Bo Miao, Mohammed Bennamoun, and Ajmal Mian are with the Department of Computer Science and Software Engineering, The University of Western Australia, Crawley, Perth, WA 6009, Australia (e-mail: bo.miao@research.uwa.edu.au; mohammed.bennamoun@uwa.edu.au; ajmal.mian@uwa.edu.au).

Yongsheng Gao is with the School of Engineering, Griffith University, Brisbane, QLD 4111, Australia (e-mail: yongsheng.gao@griffith.edu.au).

Digital Object Identifier 10.1109/TIP.2024.3381445

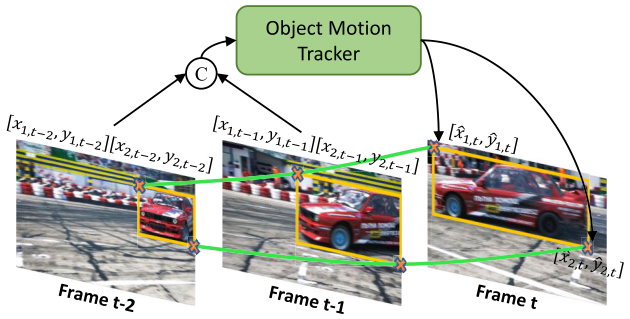


Fig. 1. Concept of our object motion tracker. Each object is tracked across frames by predicting motion functions using the position features of the object in the previous frames.

Occlusion poses challenges for matching-based VOS methods due to the similar appearances and positions of objects. Currently, no large-scale datasets are designed to evaluate semi-supervised VOS models under occlusions specifically. To fill this gap, we create a large-scale occluded video object segmentation dataset, coined OVOS, based on the OVIS dataset [20]. We further evaluate our method and recent models on OVOS to verify their ability in occlusion scenarios.

We perform extensive experiments on benchmark datasets, *i.e.*, DAVIS, YouTube-VOS, and our new OVOS dataset to evaluate the performance of our method. RAVOS achieves state-of-the-art overall performance compared to existing methods. For instance, it achieves 86.1 $\mathcal{J}\&\mathcal{F}$ with 42 FPS on DAVIS 2017 validation set, outperforming current competitors in both accuracy and inference speed. Our main contributions are summarized as follows:

- We propose motion path memory (MPM), which memorizes features within the motion path of objects to reduce redundant memory storage and accelerate feature matching and propagation.
- We propose a fast (5000 FPS) object motion tracker to track objects across frames by predicting motion functions. This enables object-level segmentation with the help of our designed object decoder.
- We create OVOS, a large-scale occluded video object segmentation benchmark, to exclusively evaluate the VOS model performance in challenging occlusion scenarios for the first time. This reveals current Semi-supervised VOS model limitations. The dataset is available at http://ieee-dataport.org/9608.
- Experiments on multiple benchmarks show that our approach achieves top-ranked performance while running twice as fast as existing ones.

II. Related Work

A. Semi-Supervised Video Object Segmentation

Semi-supervised VOS provides mask annotations for all training frames and aims to track the target objects across the entire video using (only) the first frame mask annotation during inference. Before the rise of deep learning, traditional methods often adopted graphical models [21] or optical flow [22] for video segmentation. For instance, [23] proposes a global consistency aware query strategy for VOS. Recent studies of semi-supervised VOS mainly focus on deep neural networks because of their unmatched performance.

Early deep learning-based methods often fine-tune the networks on each video during inference, making them focus on different targets [15], [24], [25], [26], [27]. For example, OSVOS [5] and its variants [6], [7], [28] fine-tune networks on the first frame or confident middle frames. Lucid Tracker [29] and PReMVOS [30] use data augmentation to generate synthetic frames for online fine-tuning. Despite their satisfying results, online fine-tuning severely limits the inference speed of networks and leads to overfitting. To accelerate VOS, DMN-AOA [14] adopts instance segmentation network to generate abundant ROIs and then perform local segmentation after non-maximum suppression operation. SAT [17] employs template matching for object localization.

To achieve better performance, recent works leverage spatio-temporal propagation [31], [32], [33], [34], [35] or pixel-wise feature matching [9], [36], [37], [38], [39], [40], [41] to guide VOS. The former propagates spatio-temporal features implicitly across frames. Among them, RVOS [42] and DyeNet [43] adopt recurrent neural networks to propagate spatio-temporal features. AGAME [44] proposes a fusion module to integrate spatio-temporal features with appearance features. The latter computes spatio-temporal correspondences for mask propagation. PML [36] adopts pixel-wise metric learning and classifies pixels based on a nearest-neighbor method. STM [8] and its variants [45], [46] memorize spatio-temporal features and perform non-local matching for temporal propagation. BATMAN [47] adopts optical flow calibration to improve smoothness and reduce noise at object boundaries. GSFM [48] proposes spectral modules to enhance intraframe interactions. XMem [49] introduces sensory memory to provide temporal locality. DeAOT [50] decouples the hierarchical propagation [10] of object-agnostic and object-specific features and performs gated propagation to efficiently achieve improved accuracy. To enable unsupervised training, MAST [12] and MAMP [13] use photometric reconstruction to learn to construct spatio-temporal correspondences. To accelerate association, RMNet [19] leverages optical flow to perform regional matching. Based on the observation that dot product affinity leads to poor memory usage, STCN [11] adopts L2 similarity for affinity measurement.

The above methods achieve good performance on semi-supervised VOS. However, they either require to segment and memorize the entire features of one frame [11], [50], which leads to redundant computations and memory storage, or rely on extra time-consuming networks, *e.g.*, optical flow, to locate ROIs. These problems restrict the deployment of VOS in memory-constrained real-time applications. Hence, an effective ROI localization and segmentation method is needed for fast and accurate VOS. We propose RAVOS, which contains an extremely fast object motion tracker to predict ROIs and employs object-level segmentation and motion path memory for efficient segmentation and memorization. Similar to the action recognition method SAOA [51], we predict ROIs to reduce target-irrelevant regions. Differently, our deep tracker employs motion functions instead of costly visual features for

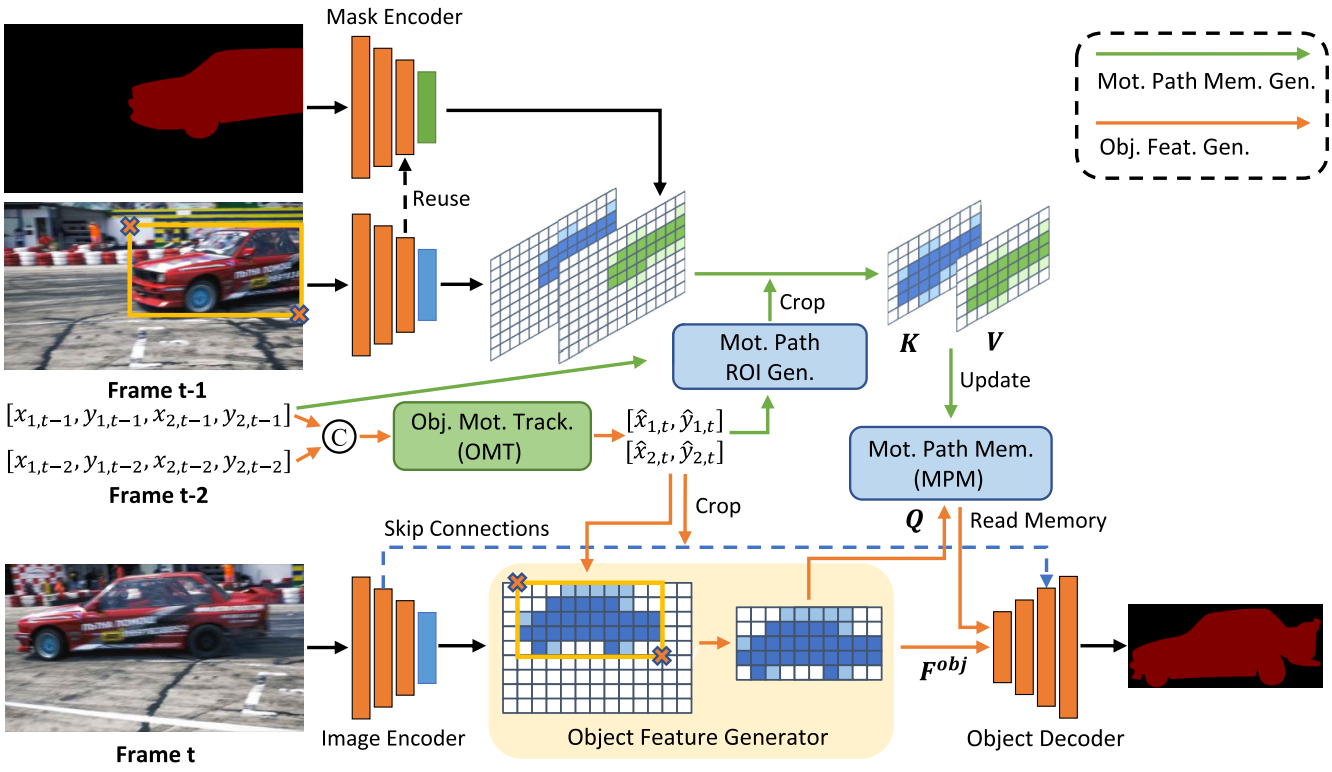


Fig. 2. RAVOS architecture. Q, K, V, and $\mathcal{F}$ denote the *query* of frame t, *key* of the memory, *value* of the memory, and appearance features of frame t, respectively. The proposed OMT estimates ROIs for target objects. Each object is then decoded and segmented at the object level according to the object ROIs, MPM, and object decoder.

temporal tracking, achieving 5000 FPS and making additional backbones and proposals unnecessary.

B. Multi-Object Tracking

Multi-object tracking (MOT) aims to continuously estimate the trajectories of target objects across frames. Object detection, association, and motion estimation are three key components of MOT. Among them, CenterTrack [52] adopts a detection network to detect object centers and predict motion offsets for tracking. TraDeS [53] estimates motion offsets to track objects, and combines the tracking results with detection results for MOT. DMMNet [54] leverages spatio-temporal features to predict tracklets for tracking. TT17 [55] proposes an iterative clustering method to generate multiple high confidence tracklets for objects. ByteTrack [56] incorporates low-confident boxes for association to dig out objects. DAN [57] proposes an affinity refinement module for more comprehensive associations.

With the help of object trackers, we can locate ROIs for region-aware segmentation and memorization. However, directly using existing MOT methods in VOS will introduce redundant architectures and computations, violating the lightweight and real-time VOS performance requirements. In this work, we propose OMT to meet the lightweight and real-time processing requirements. Instead of using image features for tracking, OMT employs the object position information in previous frames to predict the parameters of motion functions.

C. Memory Networks

Memory networks aim to capture long-term dependencies by storing temporal features or different categories of features in memory modules. LSTM [58] and GRU [59] implicitly represent spatio-temporal features with highly compressed local memory cells limiting the representation ability. Memory networks [60] were introduced to explicitly store crucial features. A classical memory-based VOS method is STM [8] which incrementally adds uncompressed features of past frames to the memory bank, and performs non-local matching to propagate spatio-temporal features. However, the background features are highly redundant. In this work, we introduce motion path memory which filters redundant context (background far from objects) while maintaining important context (foreground and nearby background).

III. Method

We propose RAVOS, an efficient and accurate semi-supervised VOS method as shown in Fig. 2. In a nutshell, RAVOS is developed based on matching-based VOS framework and contains five parts: *feature extraction*, *ROI prediction*, *memory storage*, *memory propagation*, and *object segmentation*.

RAVOS adopts ResNet-50 and ResNet-18 [61] to extract image features (*key* and *query*) and mask features (*value*) respectively. After feature extraction, we employ the proposed OMT to track objects and predict their ROIs for the current frame t, *i.e.*, $\hat{\mathcal{R}}_t \in [\hat{x}_{1,t}, \hat{y}_{1,t}, \hat{x}_{2,t}, \hat{y}_{2,t}]$. For memory updating,

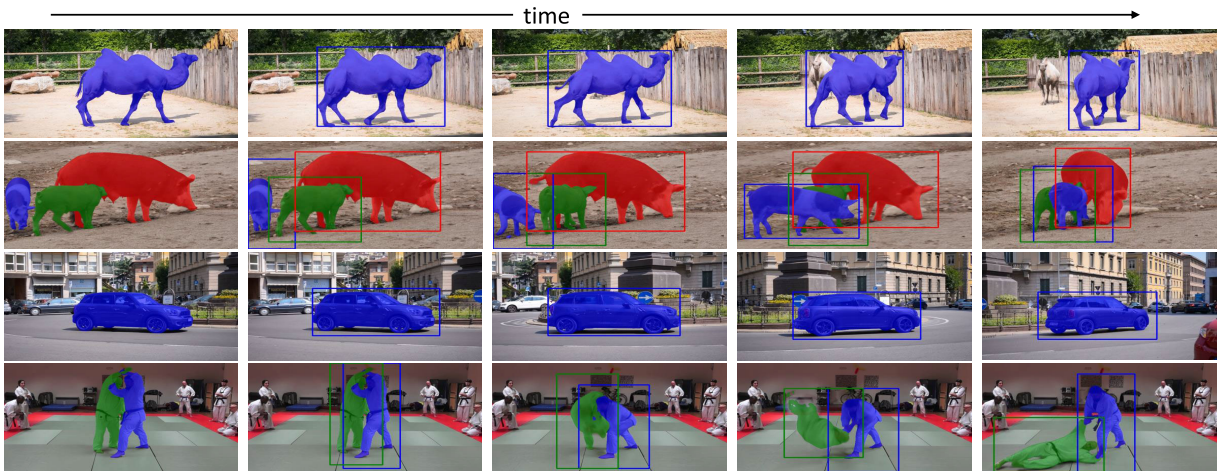


Fig. 3. Visualization of the tracking results predicted by OMT. The first column denotes the reference frames for mask propagation, and the segmentation is performed within the object ROIs.

the previous frame's object bounding boxes $\mathcal{B}_{t-1}$ and the predicted object ROIs $\hat{\mathcal{R}}_t$ are forwarded to the motion path ROI generator to generate memory ROIs in frame t-1. The MPM is then updated using features within these memory ROIs. For regional segmentation, object-level features of frame t are extracted according to the predicted object ROIs $\hat{R}_t$, and their corresponding spatio-temporal features are retrieved from the memory bank. Subsequently, an object decoder is employed to segment each object.

A. Feature Extraction

Following previous works [8], [11], we adopt ResNet-50 and ResNet-18 (excluding their last stages) [61] as the image and mask encoders, respectively. Inputs are downsampled by 1/16 via encoders. For the image encoder, one additional 3×3 convolutional layer is used on top of the *res4* features to extract *key* $K \in \mathbb{R}^{HW \times 64}$ or *query* $Q \in \mathbb{R}^{HW \times 64}$ for matching, and another 3×3 convolutional layer is leveraged on top of the *res4* features to compute appearance features $F \in \mathbb{R}^{HW \times 512}$ to assist object segmentation. Image features at the middle layers are saved to extract object skip connections for the object decoder. For the mask encoder, two residual blocks and one CBAM block [62] are used on top of the *res4* features to extract *value* $V \in \mathbb{R}^{HW \times 512}$ for each object.

B. Object Motion Tracker

An effective tracker is imperative for ROI prediction and efficient regional semi-supervised VOS. Existing deep learning-based MOT methods use appearance features for tracking. Although appearance features yield good MOT results, directly incorporating such techniques into VOS is difficult to cater for the lightweight and real-time processing requirements.

To address the problem, we propose a novel object motion tracker (OMT) that employs object position features in previous frames to predict instantaneous motion functions for MOT. As shown in Fig. 4, the deep motion estimator in OMT takes the normalized positions of an object from previous

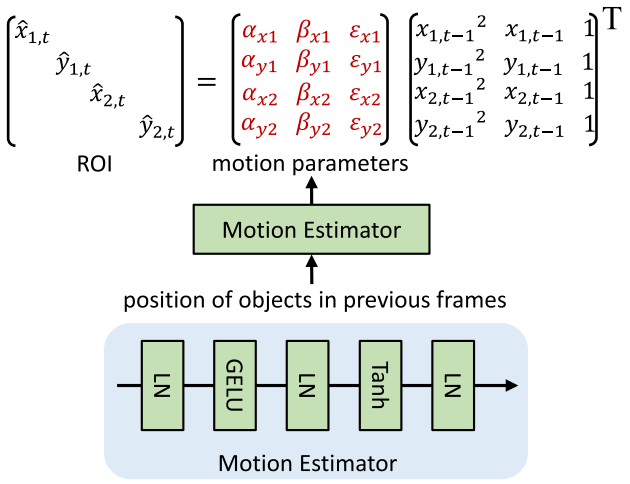


Fig. 4. Object motion tracker. The proposed tracker encodes the position of each object in previous frames into the parameters of quadratic motion functions for tracking.

frames, such as $\mathcal{B}_{t-2}$ and $\mathcal{B}_{t-1}$, and encodes them into hidden embeddings. These embeddings capture spatio-temporal details including position and scale changes to predict the parameters of motion functions for tracking. Here, $\mathcal{B}_t \in [x_{1,t}, y_{1,t}, x_{2,t}, y_{2,t}]$, representing the top-left and bottom-right corners of the bounding box and indicating the object's outermost coordinates rather than its actual corners.

In this work, we choose the quadratic function to model motion. The parameters (α, β, and ϵ) of the function are predicted by the motion estimator, and the object ROI $\hat{\mathcal{R}}_t$ in frame t is predicted by plugging $\mathcal{B}_{t-1}$ into the estimated function:

$$\begin{aligned}
\hat{x}_{1,t} &= \alpha_{x1} x_{1,t-1}{}^2 + \beta_{x1} x_{1,t-1} + \epsilon_{x1} - \phi \\
\hat{y}_{1,t} &= \alpha_{y1} y_{1,t-1}{}^2 + \beta_{y1} y_{1,t-1} + \epsilon_{y1} - \phi \\
\hat{x}_{2,t} &= \alpha_{x2} x_{2,t-1}{}^2 + \beta_{x2} x_{2,t-1} + \epsilon_{x2} + \phi \\
\hat{y}_{2,t} &= \alpha_{y2} y_{2,t-1}{}^2 + \beta_{y2} y_{2,t-1} + \epsilon_{y2} + \phi
\end{aligned} \tag{1}$$

where ϕ denotes the padding of bounding boxes to improve the segmentation robustness along the object boundary. Finally,

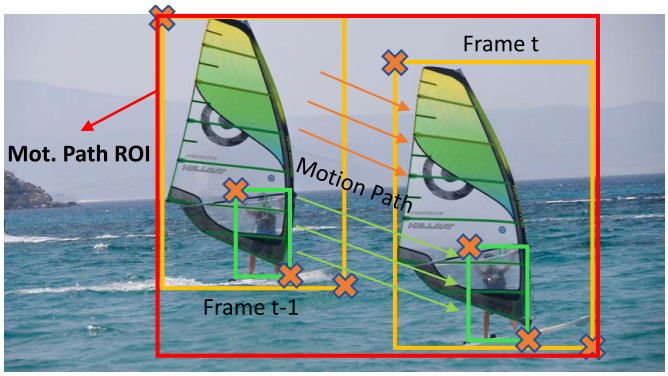


Fig. 5. Motion path ROI generation. The yellow and green bounding boxes denote the object ROIs of the sailboat and person in two frames, respectively. The red bounding box refers to the generated motion path ROI for frame t-1, which covers the motion path of all objects between two frames and filters redundant background far from objects. All bounding boxes in frame t are predicted by OMT.

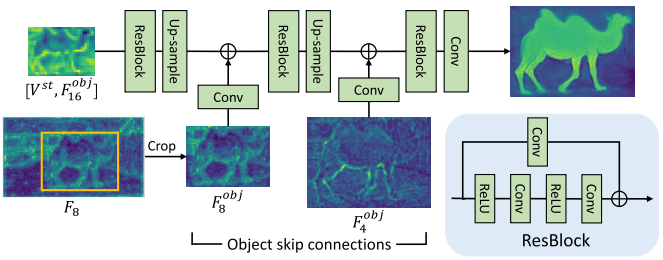


Fig. 6. Object decoder. F_{16}^{obj} and V^{st} denote the appearance features of current frame and the queried spatio-temporal features. F_8^{obj} and F_4^{obj} are object skip connections extracted from the middle layer features of the image encoder.

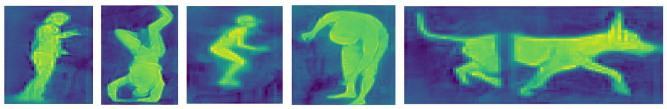


Fig. 7. Visualizations of decoded object features.

the segmentation operates within the object ROI $\hat{\mathcal{R}}_t$ to reduce computation and generate $\mathcal{B}_t$. Unlike previous template matching-based methods, OMT eliminates the need for non-local matching, which faces challenges when dealing with similar-looking objects. Different from optical flow-based methods, OMT employs lightweight but crucial spatio-temporal position information rather than resource-intensive appearance features to estimate ROIs. The lightweight framework of OMT enables it to perform at 5000 FPS on a single GPU, which is about *100*× faster than the prevalent RAFT optical flow [63]. Moreover, OMT predicts motion functions to generate a definite ROI for each object rather than generating many proposals using additional detection networks like [15] and [51] making the non-maximum suppression (NMS) operation redundant. The visualization of the tracking results shows that our efficient OMT can generate sufficiently accurate ROIs for regional VOS (see Fig. 3).

C. Motion Path Memory

Memory-based methods often accumulate local context by incrementally storing entire features from selected frames, enabling them to capture long-term dependencies and ensure temporal consistency in VOS through propagation. However, previous methods [11], [13] have shown that only a few positions in the memory are helpful for the association of a query point. Furthermore, our proposed RAVOS only segments the ROIs generated by OMT. As a result, memorizing the entire features of one frame includes redundant background context that is far from foreground objects. This redundancy hinders the efficient deployment of VOS.

For redundancy reduction, we introduce motion path memory (MPM) to memorize the critical context, *i.e.*, the foreground and nearby background. As shown in Fig. 5, MPM generates the motion path of each object between two frames and then memorizes features within the united motion paths to filter redundant context. Specifically, before memorizing the features of frame t-1, we first predict the ROIs of objects $\hat{\mathcal{R}}_t = \{\forall \hat{\mathcal{R}}_t^i, i \in [1, \ldots, N]\}$ in the next frame t using OMT. The motion path of each object between two frames is then extracted according to the position of the object in the two frames. Finally, the memory ROIs are formed by the union of all motion paths:

$$\text{ROI} = \cup(\{\forall \cup (\mathcal{R}_{t-1}^i, \hat{\mathcal{R}}_t^i), i \in [1, \ldots, N]\}) \tag{2}$$

where $\cup$ denotes the union operation and N is the number of objects. In that case, redundant background features outside the memory ROI will not be used to update the memory. Therefore, the proposed MPM not only accelerates feature matching and propagation, but also reduces the memory footprint for efficient deployment.

D. Memory Propagation

As in common matching-based VOS methods [8], [11], the memory module propagates mask features across frames for segmentation according to the pixel-wise affinity between all query and memory pixels. In this work, we perform regional matching using L2 similarity to compute the affinity.

To illustrate, we first define $Q \in \mathbb{R}^{HW\times 64}$, $K \in \mathbb{R}^{N\times 64}$, $V \in \mathbb{R}^{N\times 512}$ as the *query* of the current frame, *key* of the memory, and *value* of the memory, respectively. Where H and W are the feature height and width, and $N \ll THW$ denotes the number of positions in the memory. For each object, after predicting the object ROI $\hat{\mathcal{R}}_t$ at frame t, we crop *query* to obtain *object query* $Q^{obj} \in \mathbb{R}^{S_1 S_2\times 64}$, where S_1 and S_2 denote the height and width of $\hat{\mathcal{R}}_t$ at feature scale and $S_1 S_2 \ll HW$. Then, the affinity between Q^{obj} and K is computed as:

$$W_{i,j} = \frac{exp(\langle Q_i^{obj}, K_j\rangle)}{\sum_j exp(\langle Q_i^{obj}, K_j\rangle)} \tag{3}$$

where $\langle\cdot,\cdot\rangle$ denotes the L2 similarity between two vectors. Finally, each query position retrieves spatio-temporal features from the memory based on the computed affinity:

$$V_i^{st} = \sum_j W_{i,j} V_j \tag{4}$$

E. Object-Level Segmentation

To reduce redundant computations without losing important information, we predict object ROIs and decode object

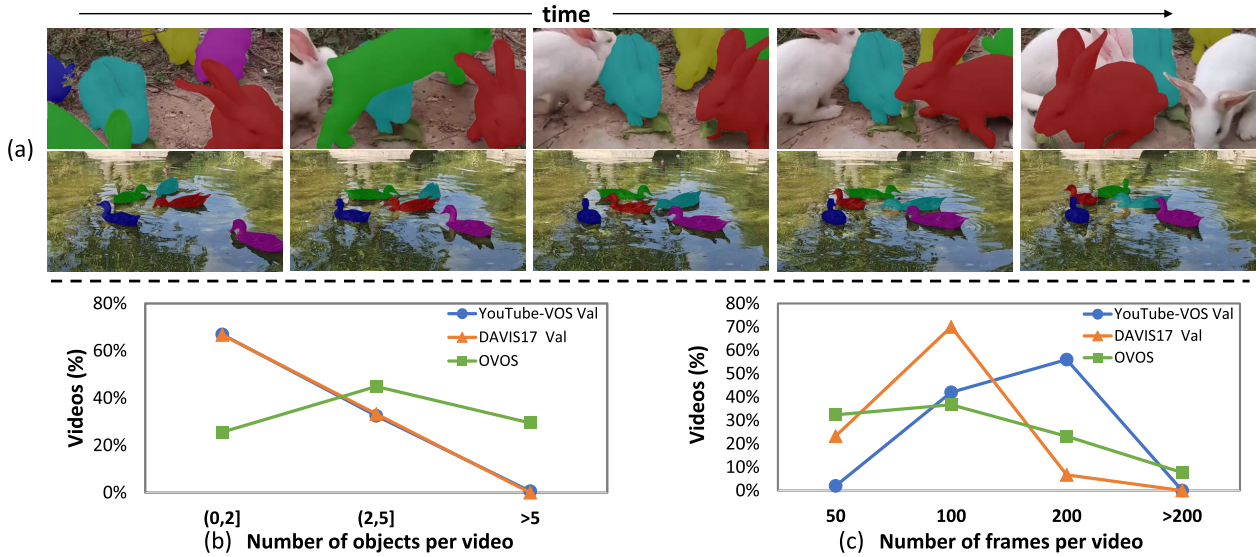


Fig. 8. Overview of the OVOS dataset. (a) Examples of video sequences. (b) Distribution of object numbers per video, where OVOS typically includes more objects in videos. (c) Distribution of frame numbers per video, where OVOS has longer videos with up to 500 frames.

features. As shown in Fig. 6, residual blocks with object skip connections are leveraged to build the object decoder. Specifically, the object decoder concatenates the object appearance features of the current frame F^{obj} and their corresponding spatio-temporal features V^{st} obtained from the memory as input, and progressively increases the object feature resolution from 1/16 to 1/4. The decoded object features are used to predict object probability maps and are up-sampled to the input resolution. Finally, the object probability maps are projected on the image probability map, and Argmax is applied for segmentation. To illustrate the effectiveness of our object decoder, we visualized the decoded object features in Fig. 7.

F. Occluded Video Object Segmentation Dataset

In this work, we introduce occluded video object segmentation (OVOS) dataset to evaluate the performance of Semi-supervised VOS models under occlusions. OVOS is an extension of the training set of OVIS dataset [20], which is proposed for video instance segmentation, since the segmentation is not available for the validation set. To align with DAVIS evaluation format, we select only objects appearing in the first frame as targets and resize videos to make their shortest size 480 pixels. The final dataset contains 607 videos with a total of 42149 frames and 2034 objects, which is larger than the current largest YouTube-VOS validation set (507 videos with a total of 13710 frames). An example is shown in Fig. 8 (a), OVOS comes with accurate annotations and includes severe object occlusions in all videos, making it ideal for testing model performance in occlusion scenarios. Moreover, as shown in Fig. 8 (b) and (c), OVOS contains a maximum of 20 objects or 500 frames within a single video, significantly increasing the challenge of occluded scenes. The dataset is available at http://ieee-dataport.org/9608.

IV. Implementation Details

A. Training

Exactly following previous works [11], [64], we pre-train the models on static image datasets [65], [66], [67], [68], [69] and perform the main training on the synthetic dataset [64] as well as DAVIS [70] and YouTube-VOS [71]. In the former stage, three synthetic frames are generated from one static image by applying random augmentation. During main training, all the video frames are resized to 480p, and three neighboring frames are randomly sampled with the maximum sampling interval ranging from 5 to 25.

In this work, all experiments are conducted in PyTorch [72] using a single 3090 GPU. Adam optimizer [73] is used to optimize the parameters. We adopt bootstrapped cross-entropy loss $\mathcal{L}_{seg}$ to train the segmentation model and mean squared error loss $\mathcal{L}_{track}$ to train the OMT:

$$\mathcal{L}_{seg} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{c} y_{i,j}\log f_j(x_i;\theta) \tag{5}$$

$$\mathcal{L}_{track} = \frac{1}{n}\sum_{i=1}^{n}(b_i - \hat{b}_i)^2, b \in [x_1, y_1, x_2, y_2] \tag{6}$$

B. Inference

We segment all videos at 480p during inference. Unless specified otherwise, RAVOS updates the memory every three frames for DAVIS and five frames for YouTube-VOS. We adopt only the top 20 matches for feature propagation as in [11]. For video segmentation, we segment entire features on the second frame and start to track objects on the third frame using the positional information of objects in the two past frames. The minimum object ROI to feature area ratio is set to 0.2 to avoid object features being too small to decode, and the padding rate is 0.15 in each direction. The object ROI is expanded to the whole image when OMT senses the disappearance of objects, and RAVOS performs regional segmentation again when the objects re-appear in subsequent frames.

V. Experiments

We evaluate RAVOS on the popular DAVIS and YouTube-VOS benchmark datasets as well as our newly created OVOS

TABLE I

QUANTITATIVE EVALUATION ON DAVIS 2016 AND 2017 VALIDATION SETS. *RS*: REGIONAL SEGMENTATION. *: RE-TIMED ON OUR MACHINE FOR FAIR COMPARISON. OUR METHOD OUTPERFORMS PREVIOUS REGIONAL SEGMENTATION VOS MODELS BY A SIGNIFICANT MARGIN

			DAVIS 2017				DAVIS 2016			
Method	RS	Backbone	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	FPS	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	FPS
FEELVOS [74] [CVPR19]	×	ResNet-101	71.5	69.1	74.0	2.0	81.7	81.1	82.2	2.2
STM [8] [ICCV19]	×	ResNet-50	81.8	79.2	84.3	19*	89.3	88.7	89.9	23*
AFB-URR [45] [NIPS20]	×	ResNet-50	74.6	73.0	76.1	4	-	-	-	-
KMN [75] [ECCV20]	×	ResNet-50	82.8	80.0	85.6	<8.3	90.5	89.5	91.5	8.3
CFBI+ [76] [PAMI21]	×	ResNet-101	82.9	80.1	85.7	5.6	89.9	88.7	91.1	5.9
SwiftNet [77] [CVPR21]	×	ResNet-50	81.1	78.3	83.9	<25	90.4	90.5	90.3	25
LCM [41] [CVPR21]	×	ResNet-50	83.5	80.5	86.5	<8.5	90.7	89.9	91.4	8.5
JOINT [78] [ICCV21]	×	ResNet-50	83.5	80.8	86.2	4	-	-	-	-
HMMN [79] [ICCV21]	×	ResNet-50	84.7	81.9	87.5	<10	90.8	89.6	92.0	10
R50-AOT [10] [NIPS21]	×	ResNet-50	84.9	82.3	87.5	24*	91.1	90.1	92.1	24*
ASRF [80] [TIP22]	×	ResNet-101	83.2	80.3	86.1	-	90.9	90.1	91.7	-
CoVOS [81] [CVPR22]	×	ResNet-50	82.4	79.7	85.1	33.7	89.1	88.5	89.6	-
RDE [82] [CVPR22]	×	ResNet-50	84.2	80.8	87.5	27.0	91.1	89.7	92.5	35.0
SWEM [83] [CVPR22]	×	ResNet-50	84.3	81.2	87.4	-	91.3	89.9	**92.6**	-
STCN [11] [NIPS21]	×	ResNet-50	85.6	82.5	88.7	20*	**91.6**	90.7	92.5	22*
FAVOS [18] [CVPR18]	✓	ResNet-101	58.2	54.6	61.8	<1	80.9	82.4	79.5	<1
FTMU [84] [CVPR20]	✓	ResNet-50	70.6	69.1	-	11.1	78.9	77.5	-	11.1
SAT [17] [CVPR20]	✓	ResNet-50	72.3	68.6	76.0	<39	83.1	82.6	83.6	39
TAN-DTTM [15] [CVPR20]	✓	ResNet-50	75.9	72.3	79.4	7.1	-	-	-	-
RMNet [19] [CVPR21]	✓	ResNet-50	83.5	81.0	86.0	<11.9	88.8	88.9	88.7	11.9
DMN-AOA [14] [ICCV21]	✓	ResNet-50	84.0	81.0	87.0	6.3	-	-	-	-
RAVOS (Ours)	✓	ResNet-50	**86.1**	**82.9**	**89.3**	**42**	**91.7**	**90.8**	**92.6**	**58**

TABLE II

EVALUATION ON DAVIS 2017 TEST-DEV SPLIT

	STM [8]	CFBI [9]	KMN [75]	RMNet [19]	GIEL [85]	R50-AOT [10]	STCN [11]	**RAVOS (Ours)**
$\mathcal{J}\&\mathcal{F}$	72.2	75.0	77.2	75.0	75.2	79.6	79.9	**80.8**
$\mathcal{J}$	69.3	71.4	74.1	71.9	72.0	75.9	76.3	**77.1**
$\mathcal{F}$	75.2	78.7	80.3	78.1	78.3	83.3	83.5	**84.5**

dataset. Region Similarity $\mathcal{J}$ (average IoU score between the segmentation and ground truth), Contour Accuracy $\mathcal{F}$ (average boundary similarity between the segmentation and ground truth), and their mean value $\mathcal{J}\&\mathcal{F}$ are used as the evaluation metrics. All results are evaluated using the official evaluation tools or servers and, unless specified otherwise, FPS is measured without automatic mixed precision.

A. Quantitative Results

DAVIS 2016 [88] is a popular single-object benchmark dataset that contains 20 videos in the validation set. For a fair comparison, we re-time the FPS for the nearest competitors on our machine, *i.e.*, STM [8], R50-AOT [10], and STCN [11]. As shown in Table I, RAVOS achieves 91.7 $\mathcal{J}\&\mathcal{F}$ with 58 FPS on DAVIS 2016 validation set, surpassing the above competitors in both accuracy and inference speed. RAVOS even outperforms STCN with significant faster inference speed (2.6× faster).

DAVIS 2017 [70] is a multi-object extension of DAVIS 2016, which contains 30 videos in the validation and test-dev split, separately. As shown in Table I, although the proposed RAVOS aims at reducing redundant segmentation and memorization, it achieves 86.1 $\mathcal{J}\&\mathcal{F}$ with 42 FPS, leading all present methods. Compared with the nearest regional segmentation competitor DMN-AOA, RAVOS achieves better performance (86.1 *vs* 84.0 $\mathcal{J}\&\mathcal{F}$) and runs more than 5× faster. Compared with the nearest competitor STCN, RAVOS surpasses it by 0.5% and runs about 2.1× faster (42 *vs* 20 FPS). Moreover, RAVOS has considerably lower FLOPs of 108G, which is only 36% of STCN (296G). We further evaluate our method on DAVIS 2017 test-dev split. As shown in Table II, RAVOS achieves 80.8 $\mathcal{J}\&\mathcal{F}$, outperforming current benchmark STCN by 0.9%.

YouTube-VOS [71] is currently the largest dataset for VOS, containing 3471 videos in the training set and 474/507 videos in the 2018/2019 validation set. YouTube-VOS splits the validation videos into seen categories and unseen categories based on whether the objects of a category appear in the training videos or not. The performance on unseen categories is used to evaluate the generalization ability of models. As shown in Table III, RAVOS achieves 84.4/84.2 $\mathcal{J}\&\mathcal{F}$ and 23/20 FPS on YouTube-VOS 2018/2019 validation set. Compared with the nearest regional segmentation competitor DMN-AOA, RAVOS outperforms it by 1.9%. Compared with the nearest competitor STCN, RAVOS has competitive performance and runs about 2× faster (23/20 *vs* 12/11 FPS on YouTube-VOS 2018/2019). Overall, RAVOS achieves state-of-the-art performance with the fastest inference time.

OVOS is a large-scale occluded VOS dataset that contains 607 videos with severe object occlusions for validation. More

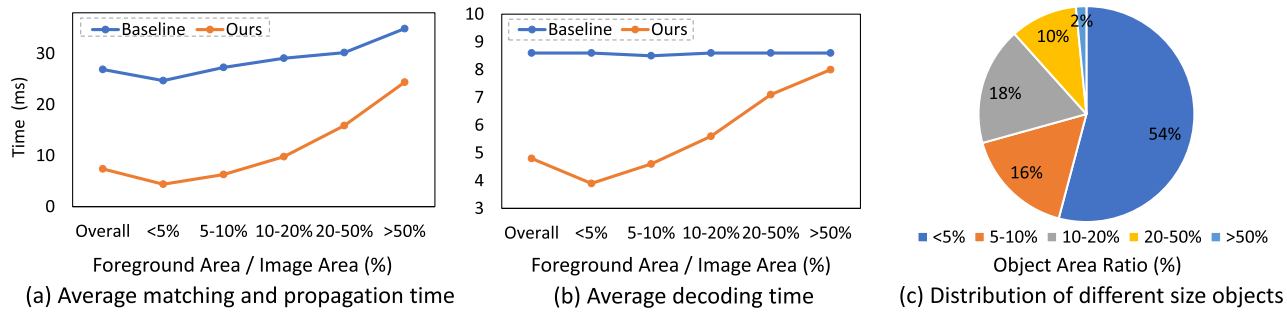


Fig. 9. Processing time for different object sizes on YouTube-VOS 2018 val. (a) Average feature matching and propagation time (ms); (b) Average feature decoding time (ms); (c) Distribution of object area ratio.

TABLE III

EVALUATION ON YOUTUBE-VOS VAL. $\mathcal{J}\&\mathcal{F}$ IS THE OVERALL PERFORMANCE ON "SEEN" AND "UNSEEN" CATEGORIES. *RS*: REGIONAL SEGMENTATION. *: RE-TIMED ON OUR MACHINE FOR FAIR COMPARISON. †: BL30K [64] IS USED FOR PRE-TRAINING

			Seen		Unseen		
Methods	RS	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	FPS
Validation 2018 Split							
STM [8]	×	79.4	79.7	84.2	72.8	80.9	-
GraphMem [86]	×	80.2	80.7	85.1	74.0	80.9	-
LWL [87]	×	81.5	80.4	84.9	76.4	84.4	-
CFBI [9]	×	81.4	81.1	85.8	75.3	83.4	3.4
CFBI+ [76]	×	82.8	81.8	86.6	77.1	85.6	4.0
SwiftNet [77]	×	77.8	77.8	81.8	72.3	79.5	-
SST [40]	×	81.7	81.2	-	76.0	-	-
GIEL [85]	×	80.6	80.7	85.0	75.0	81.9	-
ASRF [80]	×	81.9	81.0	85.8	76.3	84.3	-
LCM [41]	×	82.0	82.2	86.7	75.7	83.4	-
HMMN [79]	×	82.6	82.1	87.0	76.8	84.6	-
R50-AOT [10]	×	84.1	**83.7**	**88.5**	78.1	86.1	14.9
STCN [11]	×	83.0	81.9	86.5	77.9	85.7	12*
STCN† [11]	×	84.3	83.2	87.9	79.0	87.3	12*
SAT [17]	✓	63.6	67.1	70.2	55.3	61.7	-
TAN-DTTM [15]	✓	73.5	-	-	-	-	-
RMNet [19]	✓	81.5	82.1	85.7	75.7	82.4	-
DMN-AOA [14]	✓	82.5	82.5	86.9	76.2	84.2	-
RAVOS (Ours)	✓	83.2	82.2	86.9	77.9	85.9	**23**
RAVOS† (Ours)	✓	**84.4**	83.1	87.8	**79.1**	**87.4**	**23**
Validation 2019 Split							
STM [8]	×	79.3	79.8	83.8	73.0	80.5	-
CFBI [9]	×	81.0	80.6	85.1	75.2	83.0	3.4
CFBI+ [76]	×	82.6	81.7	86.2	77.1	85.2	4.0
SST [40]	×	81.8	80.9	-	76.6	-	-
HMMN [79]	×	82.5	81.7	77.3	86.1	85.0	-
R50-AOT [10]	×	84.1	**83.5**	**88.1**	78.4	86.3	14.9
STCN [11]	×	82.7	81.1	85.4	78.2	85.9	11*
STCN† [11]	×	**84.2**	82.6	87.0	**79.4**	**87.7**	11*
RAVOS (Ours)	✓	82.8	81.9	86.6	77.5	85.4	**20**
RAVOS† (Ours)	✓	**84.2**	82.6	87.0	**79.4**	**87.7**	**20**

details of the dataset are included in Section III-F. We directly evaluate RAVOS on OVOS dataset without retraining to verify its performance in the occlusion scenario. It is noteworthy that OVOS dataset is only used for evaluation and, to the best of our knowledge, this is the first time a semi-supervised VOS method is evaluated on this large-scale dataset. We also evaluate other successful VOS models on OVOS using a single 24GB GPU. Automatic mixed precision is used since OVOS contains some long video sequences, which cause large memory burdens and out-of-memory problems for other methods. For models still facing memory issues, we optimized their code and settings. As shown in Table IV, RAVOS outperforms STCN by 1.0% since our method performs regional segmentation to reduce the risk of false positives caused by similar-looking object occlusions. The results also expose that precisely localizing and reasoning under occlusions is still challenging for existing VOS models.

TABLE IV

EVALUATION ON OVOS USING A SINGLE 24GB GPU

Method	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
STM [8]	57.3	53.9	60.7
KMN [75]	60.3	56.6	63.9
CFBI [9]	60.7	57.4	63.9
STCN [11]	61.5	57.3	65.6
AOT [10]	62.1	**58.5**	65.7
CoVOS [81]	49.2	45.0	53.3
RDE [82]	60.0	55.5	64.4
RAVOS (Ours)	**62.5**	58.3	**66.6**

B. Ablation Studies

1) Inference Time Analysis: We first compute the distribution of object area ratio on YouTube-VOS 2018 validation set, where the size of an object is determined by the given mask of the first frame. As shown in Fig. 9 (c), nearly 70% of objects are smaller than 10% of the image area in YouTube-VOS 2018 validation set. We then analyze the single object processing time of feature matching and propagation as well as feature decoding to observe the efficiency of our method on different object sizes. As shown in Fig. 9 (a) and (b), RAVOS significantly accelerates the feature matching and propagation as well as feature decoding time on small objects. That is because the smaller the objects, the smaller is their ROIs, and the faster our method executes.

2) Object-Level Segmentation With Object Motion Tracker: Table V shows the ablation study for object-level segmentation. RAVOS outperforms baseline by 0.5% by segmenting the predicted object ROIs because of the reduced risk of false positives on background regions. Moreover, object-level segmentation significantly accelerates the feature matching time (5.9 *vs* 12.2 ms) as well as feature decoding time (3.9 *vs* 7.5 ms) due to the less redundant computations.

TABLE V
ABLATION OF MOTION PATH MEMORY (M) AND OBJECT-LEVEL SEGMENTATION (O) ON DAVIS 2017 VALIDATION SET

M	O	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	Matching (ms)	Decoding (ms)	Mem. Size (K)	FPS
×	×	85.6	82.5	88.7	12.2	7.5	36.3	20
✓	×	85.4	82.3	88.5	3.2	7.1	19.2	30
×	✓	86.1	82.9	89.3	5.9	3.9	36.3	31
✓	✓	**86.1**	**82.9**	**89.3**	**2.2**	**3.8**	**18.6**	**42**

TABLE VI
ABLATION OF TRACKERS

Tracker	$\mathcal{J}\&\mathcal{F}$	Time (ms)
Lucas–Kanade [89]	80.0	217.4
RAFT [63]	86.0	20.3
DeepSORT [90]	83.9	11.9
OMT (Ours)	**86.1**	**0.2**

TABLE VII
ABLATION OF MEMORY REGIONS

Memory	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
Foreground	85.3	82.1	88.5
Motion Path	**86.1**	**82.9**	**89.3**

TABLE VIII
ABLATION OF MOTION FUNCTIONS

Function	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
Linear	86.0	82.8	**89.3**
Quadratic	**86.1**	**82.9**	**89.3**
Cubic	86.0	82.8	89.2

TABLE IX
ABLATION OF PADDING RATES

Ratio	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
0%	85.0	81.9	88.1
5%	86.0	82.8	89.2
15%	**86.1**	**82.9**	**89.3**

TABLE X
ABLATION OF MEMORY GAPS

Method	$\mathcal{J}\&\mathcal{F}$	FPS
STCN (M=5)	85.4	24
Ours (M=5)	85.8	48
STCN (M=3)	85.6	20
Ours (M=3)	86.1	42

TABLE XI
ABLATION OF MEMORY COMPRESSION

Thred.	$\mathcal{J}\&\mathcal{F}$	FPS	Mem. Size
0.	86.1	42	18.6
0.3	**86.2**	44	14.4
0.5	85.9	47	6.3
0.7	83.1	**48**	**2.3**

3) Motion Path Memory: As shown in Table V, by memorizing the important motion path regions instead of the entire features, $\mathcal{J}\&\mathcal{F}$ drops by 0.2 on DAVIS 2017 validation set. This is because MPM filters out most of the background regions far from objects before updating the memory, leading to the loss of some prior information of backgrounds when performing global segmentation. However, leveraging MPM for object-level segmentation does not drop the performance of the model since object-level segmentation avoids segmenting these redundant background areas. Most importantly, MPM significantly reduces the feature matching time by about 3.8× (3.2 *vs* 12.2 ms) and memory size by about 1.9× (19.2 *vs* 36.3 K). This is crucial when segmenting long videos, *i.e.*, when the method is deployed on autonomous systems. In a nutshell, OMT reduces redundant segmentation to accelerate object segmentation, and MPM reduces redundant memorization to accelerate spatio-temporal feature matching and propagation. OMT and MPM are complementary, and the combination of the two modules achieves the best performance.

4) Different Trackers: We compare the performance of OMT with the traditional Lucas-Kanade optical flow [89], the cutting-edge RAFT [63], and the Kalman filter-based DeepSORT [90]. As shown in Table VI, OMT achieves superior performance in regional VOS. Most importantly, OMT only requires 0.2 ms for single object tracking, which is about 100× faster than RAFT. These results indicate that OMT is more suitable for object tracking in efficient VOS.

5) Different Memory Regions: We compare our MPM with foreground bounding boxes only memory, which does not include the motion path. As shown in Table VII, MPM outperforms foreground bounding boxes only memory by 0.8% since MPM also contains some useful background features for the ROIs in next frame.

6) Different Tracking Functions: We use different motion functions to verify the performance of OMT. As shown in Table VIII, all motion functions have good performance and the quadratic motion function obtains the best performance among the three. The results indicate the efficacy of the motion estimator in OMT in predicting the parameters of motion functions.

7) Different Padding Rates: We slightly enlarge ROIs to mitigate false negatives near the boundary. As shown in Table IX, our method maintains competitive performance even without padding. With minimal padding of 5%, the $\mathcal{J}\&\mathcal{F}$ score increases to 86.0%.

8) Different Memory Gaps: In Table X, our method consistently outperforms STCN while running 2× faster using different memory gaps, demonstrating the effectiveness of our paradigm.

Fig. 10. Tracking and segmentation results of RAVOS in severe occlusion scenario, segmentation is performed within ROIs. Our method accurately predicts ROIs and identifies target disappearance and reappearance for non-local or local segmentation.

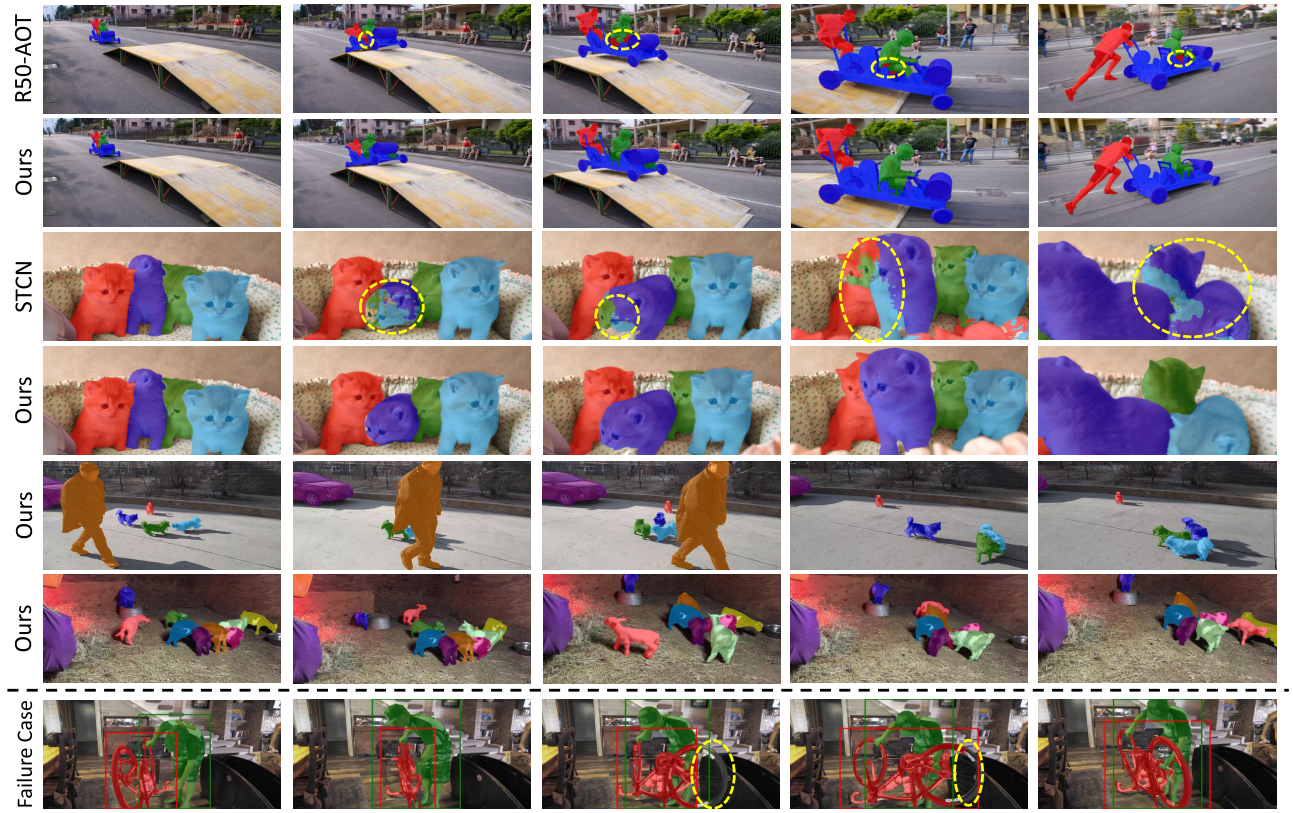


Fig. 11. (Top) Our approach predicts accurate results even in scenarios involving fast motion, similar appearances, and occlusions. (Below) Our predicted ROI initially fails to fully enclose the bicycle when it shifts to a lateral view in the third frame, due to drastic sudden scale variations. However, the predicted ROI gradually covers the entire bicycle when the change is no longer drastic.

C. Diversity-Centric Memory Compression

To further enhance efficiency, we design a diversity-centric memory compression technique. Before incorporating features F^v into memory $F^m \in \mathbb{R}^{N\times C}$, we compute the minimum relative L2 distance between each node in F^v and existing memory F^m to ensure new features enhance diversity and avoid redundancy:

$$\mathrm{Dis}(F_i^v, F^m) = min(\{\forall \frac{\left\| F_i^v - F_j^m \right\|_2}{\left\| F_j^m \right\|_2}, j \in [1, \ldots, N]\}) \quad (7)$$

Only feature nodes in F^v that exhibit a significant distance (greater than the threshold) from the memory are added. As shown in Table XI, this technique further reduces the memory size by up to 8× and increases speed by 14%, making our method more practical for processing long videos.

D. Tracking in Occlusion Scenarios

Fig. 10 visualizes some tracking and segmentation results during severe occlusions. OMT effectively captures object scale and position variations across frames to predict accurate ROIs. When targets disappear, entire features are segmented. Once the targets reappear, OMT extracts bounding boxes to resume tracking and regional segmentation.

E. Challenge of Occlusions

To further demonstrate the challenge of occlusions, we divided OVOS into a training set of 400 videos and a validation set of 207 videos. Both STCN and our model were trained from scratch using the training set. The results presented in Table XII further indicate the challenges posed by occlusions for Semi-supervised VOS models.

F. Qualitative Results

Fig. 11 (Top) shows qualitative results of RAVOS compared with R50-AOT and STCN. RAVOS performs better when

TABLE XII
EVALUATION ON SUB-OVOS VAL

Method	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
STCN	57.4	53.4	61.4
RAVOS (Ours)	58.6	54.6	62.5

multiple similar-looking objects overlap with each other since it only segments the region within ROIs for each object, thus reducing false positives.

G. Limitation

As shown in Fig. 11 (Below), the predictions made by OMT may deviate (not large enough) when objects experience drastic sudden scale variations, causing incomplete segmentation. Nonetheless, the error is not always cumulative since OMT adapts to the variations and gradually captures the complete object again when motion becomes less drastic.

VI. CONCLUSION

In this paper, we presented a novel segmentation-by-tracking approach for region aware semi-supervised VOS. Our method outperformed existing techniques on multiple benchmark datasets in accuracy with the added advantage of faster inference time. We proposed OMT which meets the requirements of fast processing and minimal redundancy to achieve a very high frame rate of 5000 FPS for object tracking and ROI prediction. On top of OMT, we designed object-level segmentation and MPM to accelerate VOS and reduce memory size by a significant margin. Moreover, we evaluated RAVOS on a newly created OVOS dataset for the first time in the community of semi-supervised VOS. We hope our RAVOS can serve as a fundamental baseline for efficient VOS and help in the advancement of research and deployment of efficient video object segmentation, video instance segmentation, and multiple object tracking.

REFERENCES

[1] T. Zhou, J. Li, S. Wang, R. Tao, and J. Shen, "MATNet: Motion-attentive transition network for zero-shot video object segmentation," *IEEE Trans. Image Process.*, vol. 29, pp. 8326–8338, 2020.

[2] B. Luo, H. Li, F. Meng, Q. Wu, and K. N. Ngan, "An unsupervised method to extract video object via complexity awareness and object local parts," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 7, pp. 1580–1594, Jul. 2018.

[3] B. Miao, M. Bennamoun, Y. Gao, and A. Mian, "Spectrum-guided multi-granularity referring video object segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 920–930.

[4] X. Zhao, Y. Pang, J. Yang, L. Zhang, and H. Lu, "Multi-source fusion and automatic predictor selection for zero-shot video object segmentation," in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 2645–2653.

[5] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool, "One-shot video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5320–5329.

[6] P. Voigtlaender and B. Leibe, "Online adaptation of convolutional neural networks for video object segmentation," in *Proc. Brit. Mach. Vis. Conf.*, 2017.

[7] T. Meinhardt and L. Leal-Taixe, "Make one-shot video object segmentation efficient again," in *Proc. NeurIPS*, 2020, pp. 10607–10619.

[8] 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.

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

[10] Z. Yang, Y. Wei, and Y. Yang, "Associating objects with transformers for video object segmentation," in *Proc. NeurIPS*, vol. 34, 2021, pp. 2491–2502.

[11] 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. NeurIPS*, vol. 34, 2021, pp. 11781–11794.

[12] 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. 6478–6487.

[13] B. Miao, M. Bennamoun, Y. Gao, and A. Mian, "Self-supervised video object segmentation by motion-aware mask propagation," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2022, pp. 1–6.

[14] S. Liang, X. Shen, J. Huang, and X.-S. Hua, "Video object segmentation with dynamic memory networks and adaptive object alignment," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 8045–8054.

[15] X. Huang, J. Xu, Y.-W. Tai, and C.-K. Tang, "Fast video object segmentation with temporal aggregation network and dynamic template matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8876–8886.

[16] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. S. Torr, "Fast online object tracking and segmentation: A unifying approach," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1328–1338.

[17] X. Chen, Z. Li, Y. Yuan, G. Yu, J. Shen, and D. Qi, "State-aware tracker for real-time video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9381–9390.

[18] J. Cheng, Y.-H. Tsai, W.-C. Hung, S. Wang, and M.-H. Yang, "Fast and accurate online video object segmentation via tracking parts," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7415–7424.

[19] 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.

[20] J. Qi et al., "Occluded video instance segmentation: A benchmark," 2021, *arXiv:2102.01558*.

[21] I. Budvytis, V. Badrinarayanan, and R. Cipolla, "Label propagation in complex video sequences using semi-supervised learning," in *Proc. Brit. Mach. Vis. Conf.*, 2010, pp. 2258–2259.

[22] L. Chen, J. Shen, W. Wang, and B. Ni, "Video object segmentation via dense trajectories," *IEEE Trans. Multimedia*, vol. 17, no. 12, pp. 2225–2234, Dec. 2015.

[23] B. Luo, H. Li, F. Meng, Q. Wu, and C. Huang, "Video object segmentation via global consistency aware query strategy," *IEEE Trans. Multimedia*, vol. 19, no. 7, pp. 1482–1493, Jul. 2017.

[24] 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. 3491–3500.

[25] H. Xiao, J. Feng, G. Lin, Y. Liu, and M. Zhang, "MoNet: Deep motion exploitation for video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1140–1148.

[26] S. Xu, D. Liu, L. Bao, W. Liu, and P. Zhou, "MHP-VOS: Multiple hypotheses propagation for video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 314–323.

[27] 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)*, 2020, pp. 7406–7415.

[28] K.-K. Maninis et al., "Video object segmentation without temporal information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 6, pp. 1515–1530, Jun. 2019.

[29] A. Khoreva, R. Benenson, E. Ilg, T. Brox, and B. Schiele, "Lucid data dreaming for video object segmentation," *Int. J. Comput. Vis.*, vol. 127, no. 9, pp. 1175–1197, Sep. 2019.

[30] 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.

[31] 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. 2186–2195.

[32] S. W. Oh, J.-Y. Lee, K. Sunkavalli, and S. J. Kim, "Fast video object segmentation by reference-guided mask propagation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7376–7385.

[33] G.-P. Ji, K. Fu, Z. Wu, D.-P. Fan, J. Shen, and L. Shao, "Full-duplex strategy for video object segmentation," in *Proc. ICCV*, 2021, pp. 4922–4933.

[34] H. Park, J. Yoo, S. Jeong, G. Venkatesh, and N. Kwak, "Learning dynamic network using a reuse gate function in semi-supervised video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8401–8410.

[35] K. Duarte, Y. Rawat, and M. Shah, "CapsuleVOS: Semi-supervised video object segmentation using capsule routing," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8479–8488.

[36] Y. Chen, J. Pont-Tuset, A. Montes, and L. Van Gool, "Blazingly fast video object segmentation with pixel-wise metric learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1189–1198.

[37] 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. 3977–3986.

[38] Y. Hu, J. Huang, and A. G. Schwing, "VideoMatch: Matching based video object segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sep. 2018, pp. 54–70.

[39] Y. Zhang, Z. Wu, H. Peng, and S. Lin, "A transductive approach for video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6947–6956.

[40] B. Duke, A. Ahmed, C. Wolf, P. Aarabi, and G. W. Taylor, "SSTVOS: Sparse spatiotemporal transformers for video object segmentation," in *Proc. CVPR*, Jun. 2021, pp. 5912–5921.

[41] L. Hu, P. Zhang, B. Zhang, P. Pan, Y. Xu, and R. Jin, "Learning position and target consistency for memory-based video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 4142–4152.

[42] C. Ventura, M. Bellver, A. Girbau, A. Salvador, F. Marques, and X. Giro-i-Nieto, "RVOS: End-to-end recurrent network for video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5277–5286.

[43] X. Li and C. C. Loy, "Video object segmentation with joint re-identification and attention-aware mask propagation," in *Proc. ECCV*, 2018, pp. 93–110.

[44] J. Johnander, M. Danelljan, E. Brissman, F. S. Khan, and M. Felsberg, "A generative appearance model for end-to-end video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8945–8954.

[45] Y. Liang, X. Li, N. Jafari, and Q. Chen, "Video object segmentation with adaptive feature bank and uncertain-region refinement," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, vol. 33, pp. 3430–3441, Dec. 2020.

[46] Y. Li, Z. Shen, and Y. Shan, "Fast video object segmentation using the global context module," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Nov. 2020, pp. 735–750.

[47] Y. Yu, J. Yuan, G. Mittal, L. Fuxin, and M. Chen, "BATMAN: Bilateral attention transformer in motion-appearance neighboring space for video object segmentation," in *Proc. ECCV*. Cham, Switzerland: Springer, 2022, pp. 612–629.

[48] Y. Liu et al., "Global spectral filter memory network for video object segmentation," in *Proc. ECCV*. Cham, Switzerland: Springer, 2022, pp. 648–665.

[49] H. K. Cheng and A. G. Schwing, "XMem: Long-term video object segmentation with an Atkinson–Shiffrin memory model," in *Proc. ECCV*. Cham, Switzerland: Springer, 2022, pp. 640–658.

[50] Z. Yang and Y. Yang, "Decoupling features in hierarchical propagation for video object segmentation," in *Proc. NeurIPS*, vol. 35, 2022, pp. 36324–36336.

[51] X. Wang, L. Zhu, Y. Wu, and Y. Yang, "Symbiotic attention for egocentric action recognition with object-centric alignment," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 6, pp. 6605–6617, Jun. 2023.

[52] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2020, pp. 474–490.

[53] J. Wu, J. Cao, L. Song, Y. Wang, M. Yang, and J. Yuan, "Track to detect and segment: An online multi-object tracker," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 12347–12356.

[54] S. Sun, N. Akhtar, X. Song, H. Song, A. Mian, and M. Shah, "Simultaneous detection and tracking with motion modelling for multiple object tracking," in *Proc. ECCV*. Cham, Switzerland: Springer, 2020, pp. 626–643.

[55] Y. Zhang et al., "Long-term tracking with deep tracklet association," *IEEE Trans. Image Process.*, vol. 29, pp. 6694–6706, 2020.

[56] Y. Zhang et al., "ByteTrack: Multi-object tracking by associating every detection box," 2021, *arXiv:2110.06864*.

[57] S. Sun, N. Akhtar, H. Song, A. S. Mian, and M. Shah, "Deep affinity network for multiple object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 104–119, Jan. 2019.

[58] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[59] K. Cho et al., "Learning phrase representations using RNN encoder–decoder for statistical machine translation," 2014, *arXiv:1406.1078*.

[60] J. Weston, S. Chopra, and A. Bordes, "Memory networks," 2014, *arXiv:1410.3916*.

[61] 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.

[62] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 3–19.

[63] Z. Teed and J. Deng, "RAFT: Recurrent all-pairs field transforms for optical flow," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Aug. 2020, pp. 402–419.

[64] H. K. Cheng, Y.-W. Tai, and C.-K. Tang, "Modular interactive video object segmentation: Interaction-to-mask, propagation and difference-aware fusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 5555–5564.

[65] L. Wang et al., "Learning to detect salient objects with image-level supervision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 136–145.

[66] J. Shi, Q. Yan, L. Xu, and J. Jia, "Hierarchical image saliency detection on extended CSSD," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 4, pp. 717–729, Apr. 2016.

[67] Y. Zeng, P. Zhang, J. Zhang, Z. Lin, and H. Lu, "Towards high-resolution salient object detection," in *Proc. ICCV*, 2019, pp. 7234–7243.

[68] H. K. Cheng, J. Chung, Y.-W. Tai, and C.-K. Tang, "CascadePSP: Toward class-agnostic and very high-resolution segmentation via global and local refinement," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8890–8899.

[69] X. Li, T. Wei, Y. P. Chen, Y.-W. Tai, and C.-K. Tang, "FSS-1000: A 1000-class dataset for few-shot segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2866–2875.

[70] 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*.

[71] N. Xu et al., "YouTube-VOS: A large-scale video object segmentation benchmark," 2018, *arXiv:1809.03327*.

[72] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. NeurIPS*, vol. 32, 2019, pp. 8026–8037.

[73] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[74] 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.*, Jun. 2019, pp. 9481–9490.

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

[76] Z. Yang, Y. Wei, and Y. Yang, "Collaborative video object segmentation by multi-scale foreground-background integration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 4701–4712, Sep. 2022.

[77] H. Wang, X. Jiang, H. Ren, Y. Hu, and S. Bai, "SwiftNet: Real-time video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1296–1305.

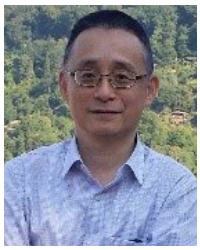
[78] Y. Mao, N. Wang, W. Zhou, and H. Li, "Joint inductive and transductive learning for video object segmentation," in *Proc. ICCV*, 2021, pp. 9670–9679.

[79] 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. 12869–12878.

[80] L. Hong, W. Zhang, L. Chen, W. Zhang, and J. Fan, "Adaptive selection of reference frames for video object segmentation," *IEEE Trans. Image Process.*, vol. 31, pp. 1057–1071, 2022.
[81] K. Xu and A. Yao, "Accelerating video object segmentation with compressed video," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 1332–1341.
[82] M. Li, L. Hu, Z. Xiong, B. Zhang, P. Pan, and D. Liu, "Recurrent dynamic embedding for video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 1322–1331.
[83] Z. Lin et al., "SWEM: Towards real-time video object segmentation with sequential weighted expectation-maximization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 1352–1362.
[84] M. Sun, J. Xiao, E. G. Lim, B. Zhang, and Y. Zhao, "Fast template matching and update for video object tracking and segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10791–10799.
[85] W. Ge, X. Lu, and J. Shen, "Video object segmentation using global and instance embedding learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 16831–16840.
[86] X. Lu, W. Wang, M. Danelljan, T. Zhou, J. Shen, and L. Van Gool, "Video object segmentation with episodic graph memory networks," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 661–679.
[87] G. Bhat et al., "Learning what to learn for video object segmentation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Nov. 2020, pp. 777–794.
[88] 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.
[89] B. D. Lucas et al., "An iterative image registration technique with an application to stereo vision," in *Proc. Image Underst. Workshop*, Vancouver, BC, Canada, 1981, pp. 121–130.
[90] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3645–3649.

Bo Miao (Graduate Student Member, IEEE) received the master's degree in signal and information processing from Jinan University. He is currently pursuing the Ph.D. degree with the School of Computer Science and Software Engineering, The University of Western Australia, under the supervision of Prof. Ajmal Mian and Prof. Mohammed Bennamoun. His research interests include video segmentation and computer vision.

Mohammed Bennamoun (Senior Member, IEEE) is currently a Winthrop Professor with the Department of Computer Science and Software Engineering, The University of Western Australia (UWA), and a Researcher of computer vision, machine/deep learning, robotics, and signal/speech processing. He has published four books (available on Amazon), one edited book, one Encyclopedia article, 14 book chapters, more than 200 journal articles, more than 250 conference publications, and 16 invited and keynote publications. His H-index is 74 and his number of citations is more than 28,000 (Google Scholar). He was awarded more than 80 competitive research grants (approximately more than $35 million in funding) from the Australian Research Council and numerous other government, UWA, and industry research grants. He has delivered conference tutorials at major conferences, including the IEEE Computer Vision and Pattern Recognition (CVPR 2016), the 2014 Interspeech, the IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), and European Conference on Computer Vision (ECCV). He served for two terms (three years each term) on the Australian Research Council (ARC) College of Experts and the 2018 ARC Excellence in Research for Australia (ERA). He is currently a Senior Area Editor of IEEE SIGNAL PROCESSING LETTERS and an Associate Editor of IEEE TRANSACTIONS ON IMAGE PROCESSING and IEEE TRANSACTIONS ON ARTIFICIAL INTELLIGENCE.



Yongsheng Gao (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in electronic engineering from Zhejiang University, Hangzhou, China, in 1985 and 1988, respectively, and the Ph.D. degree in computer engineering from Nanyang Technological University, Singapore. He was the Leader of Biosecurity Group, Queensland Research Laboratory, National ICT Australia (ARC Centre of Excellence); a Consultant of Panasonic Singapore Laboratories; and an Assistant Professor with the School of Computer Engineering, Nanyang Technological University. He is currently a Professor with the School of Engineering and Built Environment, Griffith University, and the Director of the ARC Research Hub for Driving Farming Productivity and Disease Prevention, Australia. His research interests include smart farming, machine vision for agriculture, biosecurity, face recognition, biometrics, image retrieval, computer vision, pattern recognition, environmental informatics, and medical imaging.

Ajmal Mian (Senior Member, IEEE) is currently a Professor of computer science with The University of Western Australia. His research interests include 3D computer vision, machine learning, and video analysis. He is an IAPR Fellow and a Distinguished Speaker of ACM. He was a recipient of three esteemed fellowships from the Australian Research Council (ARC). He has also received several research grants from ARC, the National Health and Medical Research Council of Australia, the U.S. Department of Defense, and Australian Department of Defense with a combined funding of over $41 million. He received the West Australian Early Career Scientist of the Year 2012 Award and the HBF Mid-Career Scientist of the Year 2022 Award. He has also received several other awards, including the Excellence in Research Supervision Award, the EH Thompson Award, the ASPIRE Professional Development Award, the Vice-Chancellors Mid-Career Award, the Outstanding Young Investigator Award, and Australasian Distinguished Dissertation Award. He was the General Co-Chair of DICTA in 2019 and ACCV in 2018. He served as a Senior Editor for IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS and an Associate Editor for IEEE TRANSACTIONS ON IMAGE PROCESSING and *Pattern Recognition* Journal.