Object Segmentation-Assisted Inter Prediction for Versatile Video Coding

Zhuoyuan Li, *Graduate Student Member, IEEE,* Zikun Yuan, Li Li, *Member, IEEE,*
Dong Liu, *Senior Member, IEEE,* Xiaohu Tang, *Senior Member, IEEE,* and Feng Wu, *Fellow, IEEE*

Abstract—In modern video coding standards, block-based inter prediction is widely adopted, which brings high compression efficiency. However, in natural videos, there are usually multiple moving objects of arbitrary shapes, resulting in complex motion fields that are difficult to represent compactly. This problem has been tackled by more flexible block partitioning methods in the Versatile Video Coding (VVC) standard, but the more flexible partitions require more overhead bits to signal and still cannot be made arbitrarily shaped. To address this limitation, we propose an object segmentation-assisted inter prediction method (SAIP), where objects in the reference frames are segmented by some advanced technologies. With a proper indication, the object segmentation mask is translated from the reference frame to the current frame as the arbitrary-shaped partition of different regions without any extra signal. Using the segmentation mask, motion compensation is separately performed for different regions, achieving higher prediction accuracy. The segmentation mask is further used to code the motion vectors of different regions more efficiently. Moreover, the segmentation mask is considered in the joint rate-distortion optimization for motion estimation and partition estimation to derive the motion vector of different regions and partition more accurately. The proposed method is implemented into the VVC reference software, VTM version 12.0. Experimental results show that the proposed method achieves up to 1.98%, 1.14%, 0.79%, and on average 0.82%, 0.49%, 0.37% BD-rate reduction for common test sequences, under the Low-delay P, Low-delay B, and Random Access configurations, respectively.

Index Terms—Inter prediction, motion compensation, motion estimation, motion vector coding, object segmentation, partition estimation, video coding, VVC.

I. INTRODUCTION

Block-based hybrid coding framework has been widely adopted in the modern video coding standards, including H.265/High Efficiency Video Coding (HEVC) [1] and the state-of-the-art Versatile Video Coding (VVC) [2]. To reduce the temporal redundancy in sequential frames, inter prediction makes a major contribution to these standards and plays a key role in hybrid video coding schemes [3]–[6].

In general, inter prediction is used to predict the current frame from previously coded frames. To improve the coding performance and reduce the implementation difficulty, block-based inter prediction is used to generate the predictive frame. In block-based inter prediction, the current frame is usually divided into non-overlapping rectangular blocks as the basic units to match prediction. For each block, a motion vector (MV) is used to indicate the displacement between the to-be-coded block and the prediction block in the reference frame. In the entire prediction process, the MV of each block is derived by motion estimation (ME). Then the block-based motion compensation (MC) is utilized to generate the prediction blocks by the indication of MV.

Block-based ME and MC are generally performed on the assumption that the motion of pixels within a block tends to be uniform. This assumption makes the motion field easy to represent with low complexity in hybrid coding framework. However, in natural videos, there are usually multiple moving objects of arbitrary shapes in video frames, leading to complex motion fields. When using the block-based motion information description method, the block-level MV may not reflect the pixel-level actual motion, and significant prediction errors may be incurred. Therefore, obtaining the flexible partition for moving objects and compactly representing their motion fields is the key to reducing the prediction errors, thereby achieving a good trade-off between the bit consumption of motion information and prediction accuracy. During the development of video coding standards, many partitioning methods have been proposed to approach the efficient representation of complex motion fields. These methods can be classified into three categories: rectangular partition methods, line-based geometric partition methods, and segmentation-based partition methods.

For rectangular partition (RP) methods, as shown in Fig. 1 (a), the rectangular partition structure [7]–[9] is widely used in block-based video coding standards. Partitioning blocks into smaller ones can handle complex motions to some extent. However, due to the inherent shape limitation of the block-based partition, it is difficult to align with natural objects of arbitrary shape. In particular, the block containing object boundary usually consists of regions in different motion fields. If the rectangular partition is used to capture the real motion of each region, it may cost a number of bits.

Line-based geometric partition (GP) methods, as shown in Fig. 1 (b), have been intensively studied to capture the non-uniform motion field in HEVC [10] and VVC [11]. In GP, a predefined straight line is used to split a rectangular block into two wedge-shaped sub-regions. Each sub-region has its motion information and performs MC to obtain its prediction. Although the geometric partition can divide the motion-

Z. Li, L. Li, D. Liu, and F. Wu are with the CAS Key Laboratory of Technology in Geo-Spatial Information Processing and Application System, University of Science and Technology of China, Hefei 230027, China (e-mail: zhuoyuanli@mail.ustc.edu.cn; lil1@ustc.edu.cn; dongeliu@ustc.edu.cn; fengwu@ustc.edu.cn).
Z. Yuan and X. Tang are with the Information Security and National Computing Grid Laboratory, Southwest Jiaotong University, Chengdu 610031, China (e-mail: yuanzikun@my.swjtu.edu.cn; xhutang@swjtu.edu.cn).

(a) Rectangular partition (RP)   (b) Line-based geometric partition (GP)   (c) Segmentation-based partition (SP)
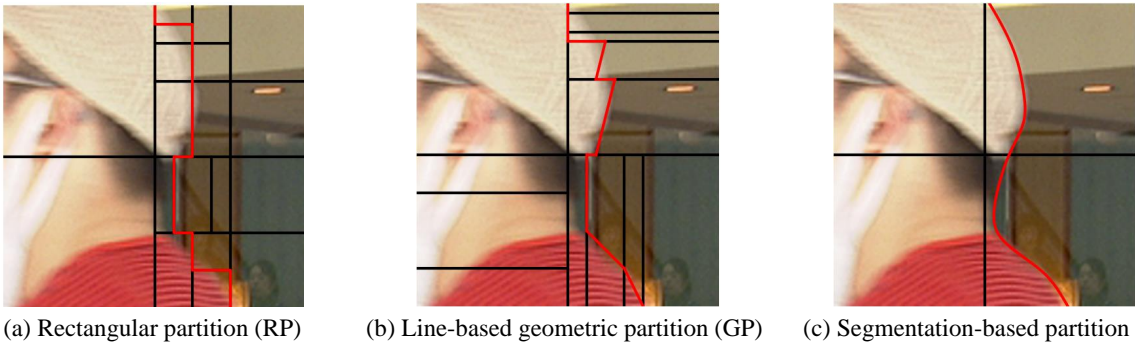
Fig. 1. The partition results of different methods in the actual coding process, where (a) uses the rectangular partition (RP), (b) uses the RP + line-based geometric partition (GP), (c) uses the RP + segmentation-based partition (SP). The block is in the 71-th frame of the *BQMall* sequence.

inconsistent sub-regions in a block to individually perform MC, the straight line can only approximate the discontinuities of the motion field. Meanwhile, extra bit consumption for partition line representation is unavoidable.

Segmentation-based partition (SP) methods, as shown in Fig. 1(c), tentatively introduce the segmentation mask as the partition representation into the existing video coding standards. [12]–[14] proposed the irregular segmented line to accurately approximate the arbitrary object shape by applying the object-oriented segmentation algorithm on the reference frame/block. The same segmentation algorithm is executed at the encoder and decoder to reduce the overhead bits for the transmission of partition information. Although the related work of SP achieves good results to some extent, they still have two drawbacks. First, previous segmentation algorithms mostly rely on hand-crafted constraints, and it is difficult to approximate the accurate partition of different motion-inconsistent regions. Second, with the segmentation obtained, previous SP methods do not sufficiently take advantage of the object-aware partition information hints for coding process, which limits the potential of fine-grained partition to improve the coding efficiency.

Recently, with the repaid development of segmentation technologies [15]–[18], moving objects can be pixel-accurately segmented in video frames. Inspired by them, we rethink the assistance of segmentation applied in inter prediction, and sufficiently explore the guidance of accurate object-aware partition information for prediction and coding process in codec. In this paper, we propose an object segmentation-assisted inter prediction (SAIP) method that does not restrict the partitioning shape and further takes advantage of the segmentation information to assist the entire inter prediction process, including MC, motion vector coding (MVC), and ME.

Specifically, in our method, some advanced segmentation technologies are used to segment objects in the reference frames. With a proper indication, the object segmentation mask is translated from the reference block to the coding block as the arbitrary-shaped partition of motion-inconsistent regions. Note that the partition information can be inferred based on the coded motion information instead of being explicitly signaled. With the assistance of the translated segmentation mask, integer-pixel and fractional-pixel MC are separately performed for different regions to achieve higher prediction accuracy. In addition, the segmentation mask is utilized to assist the MVC of different regions by adaptively selecting

and ranking the high-probability candidates to construct the candidate list, which considers different qualities of the potential MV candidates to improve the efficiency of motion data transmission. In particular, the segmentation mask is applied to the joint rate-distortion optimization (RDO) for accurate partition estimation and ME at the encoder. The derivation of different MV candidates in ME implicitly considers the various partition candidates in partition estimation for the coding block to estimate jointly.

In summary, we have made the following contributions that will be detailed in this paper:

- We propose an object segmentation-assisted inter prediction framework to leverage the segmentation to assist the entire inter prediction process, including MC, MVC, and ME. In this framework, we test the recently advanced segmentation method and verify its effectiveness for video coding.
- We design a segmentation-assisted motion compensation (SA-MC) method to separately compensate for the different motion-inconsistent regions with specific design of different MC steps to achieve accurate region-level MC.
- We design a segmentation-assisted motion vector coding (SA-MVC) method to efficiently code the MVs of different motion-inconsistent regions.
- We design a segmentation-assisted rate-distortion optimization (SA-RDO) to joint the partition estimation and ME, and further precisely estimate the partition of coding block and the MV of different motion-inconsistent regions.

II. RELATED WORK

In this section, we review the previous work that relates to our research in two aspects. First, we introduce some methods for tackling the representation of complex motion fields in video coding standards. This problem can be viewed as a partition problem, and these methods can be further divided into three categories: rectangular partitioning schemes, line-based geometric partitioning schemes, and segmentation-based partitioning schemes. Second, we introduce some advanced segmentation technologies.

A. Partitioning Schemes

1) Rectangular Partitioning Schemes: Dividing blocks into smaller ones helps the codec achieve a more accurate motion description. Rectangular partitioning structures are widely used in H.265/HEVC and H.266/VVC, which significantly

influence the compression performance. In HEVC, a quadtree (QT) partition structure [8] is adopted to replace the macroblock used in previous H.264/AVC [7]. Using the quadtree structure is beneficial for adapting to various local content in natural videos. In VVC, a quadtree plus nested multi-type tree (QTMTT) structure is further introduced, which enables binary-tree (BT), and ternary-tree (TT) partitions besides QT partition [9]. The QTMTT structure possesses more flexibility because it can generate coding units with more shapes and sizes. Although the coding performance benefits from the flexibility of partition structure, it is difficult to represent the partition of motion area for natural objects with arbitrary shapes. Particularly, the blocks with object boundaries contain multiple sub-regions moving in different directions. With the block-based motion assumption, rectangular partitioning is utilized to capture the real motion characteristics for each region, which will cause more overhead bits.

2) Line-Based Geometric Partitioning Schemes: Line-based geometric partitioning schemes (GEO) have been intensively studied to capture the non-uniform motion field [10], [11], [19], [20] during the development of HEVC and VVC. In GEO [11], [19], a predefined straight line is used to split the rectangular block into two wedge-shaped sub-parts. Each sub-part has its MV to describe the motion field and performs MC to obtain the prediction. In VVC, merge-mode-based GEO with 64 partitioning modes has been adopted [11]. The mode is selected for a coding unit from the predefined 64 partition line candidates, and the description information of partition line is represented in the form of the offset parameter ρ and angle parameter φ. For motion information of sub-parts, only unidirectional motion information is used, and the GEO merge list is derivated from the regular merge list by the parity of the GEO merge index. To economize the bits consumption of GEO, [20] proposed to utilize the spatio-temporal correlation to guide the efficient partitioning mode derivation and motion information coding. Although these methods [10], [11], [19], [20] can divide the motion-inconsistent sub-parts and represent their motion information with limited bits consumption in a block to perform MC, the straight line can only approximate the discontinuities of the motion field. Meanwhile, bit consumption for partition line representation is unavoidable.

3) Segmentation-Based Partitioning Schemes: To represent the motion field of arbitrary-shaped object accurately in natural videos, segmentation-based partitioning and coding schemes are tentatively introduced into video coding frameworks and standards [21]–[24]. In the segmentation-based partitioning schemes, the partition is mainly derived from the reference frame/block by applying a segmentation algorithm at the encoder/decoder.

In [25], Pardas *et al.* proposed a segmentation-based partition tree theory for low-bit coding. The partition of multiple regions is derived from the projected merging and segmentation. However, the region-based partition is a rough estimation for the high-texture regions, and how to efficiently extract and represent the irregular partition line information needs to be further studied. In [26], the predictive motion field segmentation method was proposed to tackle the problem of

partition signaling. To reduce the overhead bits for partition information, [12], [13] proposed that the object-oriented partition information was derived from the reference frame by the conventional segmentation method, and one MV was used to signal the position where the segmentation matched the object edges. In [27], Kim *et al.* proposed a novel segmentation method, in which two MVs were signaled and the differences between these two prediction blocks were used for segmentation. Although these methods [12], [13], [27] adopt different strategies to derivate and signal the partition information, the overhead bits are unavoidable. Meanwhile, for ME of different regions, most methods are directly inherited from the normal prediction modes (like *AMVP* and *Merge*) without any modification, and more overhead bits for motion information coding are needed.

Based on these limits, in [14], Wang *et al.* proposed a three-zone segmentation-based partitioning method to achieve object-oriented MC for prediction refinement. Based on zone partitioning, different MC methods are used for various zones. However, to reduce the time complexity in partition derivation, [14] derivated the segmentation of the optimal prediction block obtained by all inter prediction modes as the partition of current block. And the motion information of each zone is derivated under the strong assumption that the foreground zone's motion is intense and the background zone's motion is relatively slight. It may not essentially tackle the problem of multiple motion information matching for different zones in a block. And the hypothesis-based partition and MV derivation are individually performed, which limits the accuracy of partition and MV, thereby affecting the prediction quality.

B. Advanced Segmentation Technologies

1) Image Instance Segmentation (IIS): Image segmentation is an essential component in many visual understanding systems, which aims to detect all objects in the input image and assign a pixel-level mask with a category label for each instance of interest in the image [15]. It involves partitioning images into multiple segments or objects. Over the past few years, deep learning models have yielded a new generation of image segmentation methods with remarkable performance improvements on popular benchmarks. As the most classic and effective IIS scheme, Mask R-CNN [28] introduced a fully convolutional mask head to the Faster R-CNN [29] detector. Cascade Mask R-CNN [30] combined Cascade R-CNN with Mask R-CNN to achieve good segmentation results. Recently, the Sample Consistency Network (SCNet) [31] proposed a novel training method and reinforced the reciprocal relationships among subtasks, whose average precision (AP) for box and mask prediction was 48.3 and 42.7, respectively. The running time is 38% faster than the Cascade Mask R-CNN.

2) Video Object Segmentation (VOS): Video Object Segmentation is an emerging high-level video processing task and has been extensively studied in recent years. It refers to identifying and segmenting the dominant, general objects in video sequences. It is widely used in video analysis and editing-related application scenarios. Recently, most methods fit a model using the initial segmentation [32] or leverage temporal propagation [33], [34], particularly with spatio-
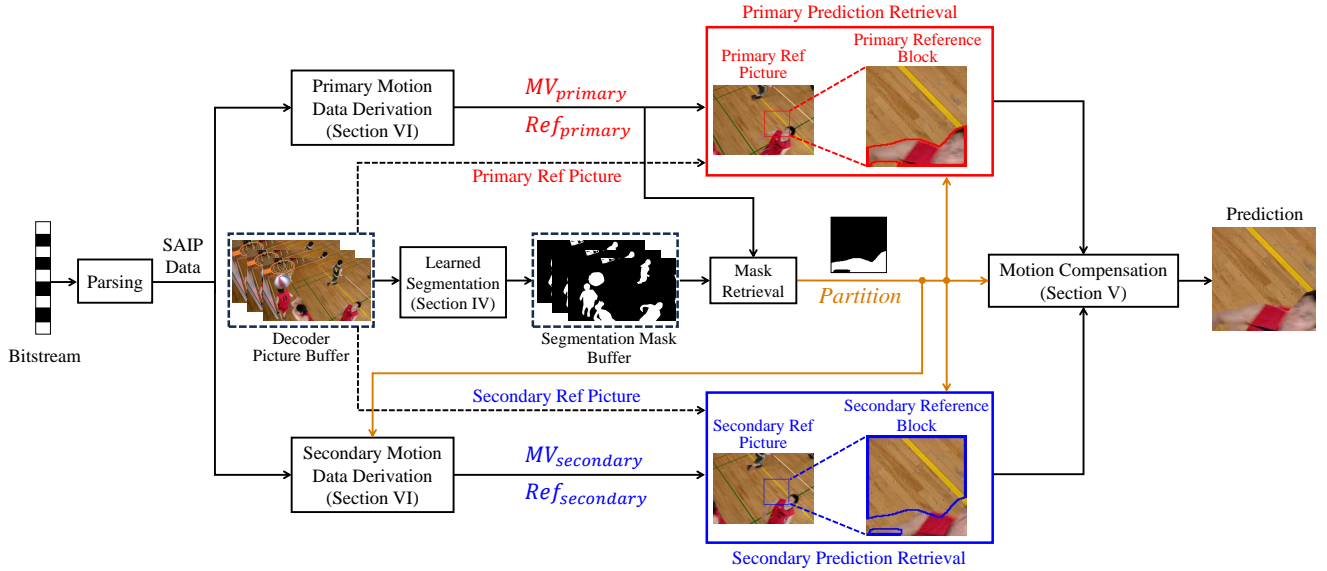
Fig. 2. Illustration of the proposed object segmentation-assisted inter prediction (SAIP) framework from the perspective of decoding. Boxes represent the sub-modules of SAIP, and arrows indicate the data flow direction. Section indicates the corresponding chapter of the module.

temporal matching [35]. Space-time memory-based networks (STM) [36] are popular due to their high performance and simplicity. As the best-performing VOS in recent years, space-time correspondence network (STCN) [37] subtracts from STM to arrive at a minimalistic form of matching networks and achieves new state-of-the-art results on the benchmark [38].

III. PROPOSED FRAMEWORK

Figure 2 shows our proposed object segmentation-assisted inter prediction (SAIP) framework from the perspective of decoding. In SAIP, the key lies in that the segmentation information is utilized to assist the overall prediction and coding process in codec. It mainly consists of four modules, including MC (Section V), MVC (Section VI), rate-distortion optimization for ME and partition estimation (Section VII), segmentation for reference pictures (Section IV). The difference between the encoder and decoder is only the additional process of ME and partition estimation. In this section, we briefly introduce the process of SAIP in the decoder.

In the decoding process (Fig. 2), different from the previous inter prediction methods, the coding block is divided into multiple motion-inconsistent regions (primary region, secondary region) to match the prediction in SAIP. First, the motion data of the current coding block is obtained by parsing the bitstream. Concerning the motion data for the coding block, two MVs ($MV_{primary}$, $MV_{secondary}$) and reference picture indexes ($Ref_{primary}$, $Ref_{secondary}$) correspond to the motion of different regions (primary motion, secondary motion). The partition of the coding block is translated from the segmentation mask of the reference block by the guidance of $MV_{primary}$. Second, the segmentation mask is utilized to derive the motion information of the secondary region ($MV_{secondary}$). Third, with the assistance of the segmentation mask, the final prediction can be derived by MC based on MVs of different regions.

IV. LEARNED SEGMENTATION FOR REFERENCE PICTURES

Obtaining the accurate partition of the motion area is the key to achieving efficient inter prediction. In the traditional

coding process, the block-based partition is difficult to align with the irregular shape of the natural object. With the maturity of deep learning-based segmentation methods, the pixel-level demarcation representation ability described by the segmentation map is also increasing. Therefore, the combination of the segmentation method with pixel-level demarcation representation precision and the codec with block-level precision has the potential to more accurately characterize video motions along the time, thus further improving the efficiency of video coding.

Video object segmentation (VOS) aims to identify and segment target instances in a video sequence. According to the strong temporal correlation between multiple video frames, the VOS model mainly propagates the object target information of the label frame to infer the segmentation for the remaining frames. For each frame, the mask represents the segmentation information of the foreground and background. In our work, VOS is utilized to assist the partition process in obtaining the accurate partition of every object's shape. We introduce the recent state-of-the-art segmentation method [31] [37] into codec to derive the object segmentation of the multiple reconstructed reference frames and translate the segmentation map of the reference frame to the current frame as the partition of motion-inconsistent regions. Here we focus on the details of how to embed the segmentation module into the codec, and the detailed network design follows the configurations in [31] [37].

The VOS methods can be grouped into two categories in common scenarios: unsupervised and semi-supervised. The difference is that the latter needs to obtain the annotation of the first frame as the tracking label source before the segmentation process. Due to the coding process, the segmentation process needs to be fully automatic. Although some automatic VOS methods can be automated, their segmented precision is far less accurate than that of semi-supervised VOS (SVOS). Therefore, we apply the state-of-the-art SVOS model space-time correspondence network (STCN) [37] to our framework. The STCN is a hybrid propagation-based and detection-based method based on a spatio-temporal attention mecha-
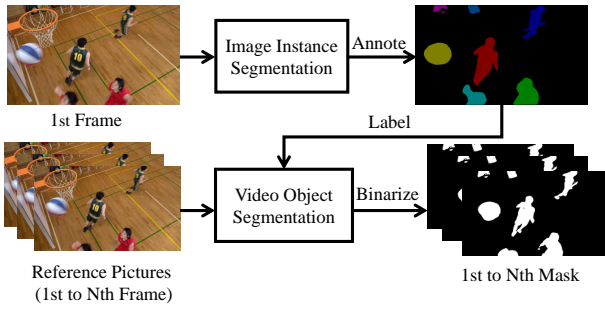
Fig. 3. The object segmentation workflow on the reference frames.

nism, which models the complex dynamic spatio-temporal dependencies of instances in multiple frames and excellently captures the motion area to achieve accurate segmentation of instances in each frame. To replace the process of manually extracting the first frame annotations in SVOS with automatic network generation, we propose cooperation between SVOS and image instance segmentation (IIS). IIS is utilized to obtain the annotations required for the first frame of SVOS in the coding process. For IIS, we apply the state-of-the-art IIS model sample consistency network (SCNet) [31].

The derivation process of the reference picture's segmentation map is shown in Fig. 3 . From the experiment for different segmentation configurations in the reference pictures, we find that the segmentation accuracy of IIS is influenced by the quantization parameters (QP) of the reference picture, and the VOS is not much affected by the picture quality. To fit the coding frame structure (group of pictures, GOP), SCNet is utilized to segment the first frame of the GOP as the tracking label source of SVOS and the STCN segments the other frames of the GOP.

When dealing with multi-frame segmentation in a GOP, first, SCNet is utilized to detect and segment foreground instances in the first frame of a GOP and obtain the info-graphic of instances, which contains the bounding box and segmentation of instances. Second, the instance segmentation information of the first frame S_0 is used as the reference for STCN inference. According to S_0, the STCN tracks the instance in each subsequent frame and obtains S_i of the subsequent frame. To represent different instances in the mask, the segmentation of each frame S_i uses different colors to represent different instances and the mask M_i of each frame is derived by binarizing S_i. Finally, we add a buffer memory for each frame to store all M_i to facilitate data interaction. The derivation of M_i is expressed as follows:

$$M_i(x,y) = \begin{cases} 1 & \text{if } S_i(x,y) \in \Phi_I \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where $M_i(x,y)$ and $S_i(x,y)$ represent the pixel values of M_i and S_i, and Φ_I denotes the instance sets.

In the case of coding requirements, sequences are also long-term. VOS methods are especially suitable for processing short-term sequences. Although the STCN has largely solved the common challenges of VOS such as object occlusion, fast motion and deformation to a certain extent, STCN still cannot solve more complex motion very well in long-term sequences. We find many scene changes in long-term sequences with new objects appearing and camera movement. Since the STCN

does not have the segmentation information of these new objects in the annotation of the first frame, it cannot segment such new objects in time, resulting in a decrease in the accuracy of the segmentation area and contour of the frame. To solve this problem, the interval of the label mask's update can be adjusted (such as 1st-GOP, 2nd-GOP, 3rd-GOP) to adapt to the scene change of different videos. At the beginning of every segmentation interval, SCNet re-derives the object information and re-propagates the mask with the STCN, which is equivalent to updating the object information regularly to guide the accurate segmentation of SVOS.

V. SEGMENTATION-ASSISTED MOTION COMPENSATION

As mentioned in Section IV, the reference frame can be segmented with the same setup in the encoder and decoder. The segmentation mask of the reference frame is generated when the coding frame is reconstructed, and stored in buffer memory for segmentation-assisted inter prediction of subsequent frames. With a proper indication, the object segmentation mask of the reference block is translated to the coding block as the arbitrary-shaped partition of motion-inconsistent regions. In this section, we assume that the partition and the motion information of different regions are obtained, and introduce how to utilize the segmentation mask to assist the MC for different motion-inconsistent regions (sub-regions) within the block.

For different sub-regions, two MVs ($MV_{primary}$ and $MV_{secondary}$) are used to represent their motion and correspond to two block-level reference blocks named primary prediction and secondary prediction ($P_{primary}$, $P_{secondary}$). Considering the accurate representation of motion information, we propose segmentation-assisted motion compensation (SA-MC) to generate the prediction of different sub-regions within a block, the process of SA-MC is shown in Fig. 4 . In particular, the segmentation mask assists the two steps of MC, including the integer and fractional parts.

A. Segmentation-Assisted Integer-Pixel Motion Compensation

In regular integer-pixel MC (IMC), the prediction can be directly copied from the reference frame. In SAIP, if the prediction of each sub-region is directly copied from different reference blocks and joined into a block to generate the block-level prediction, the prediction may result in serious boundary artifacts, thus affecting prediction accuracy. Inspired by overlapped block motion compensation [39], [40], we design an overlapped region-based motion compensation (ORMC) method, which uses the arbitrary-shaped (non-linear) partition of motion area and multiple reference blocks to generate the prediction without the region-level boundary artifacts.

On both sides of the non-linear partition line, the 2-pixel range region is defined as the edge region. For the edge region to be predicted, not only its own MV but also the MV of the neighboring region can be used to derive the prediction signal. Fig. 5 shows a set of ORMC examples. A blending process is applied to the pixels around the non-linear partition line by a distance-based adaptive weighted strategy. Pixels are classified into three categories: one pixel away from the partition line, two pixels away, and more than two pixels away.
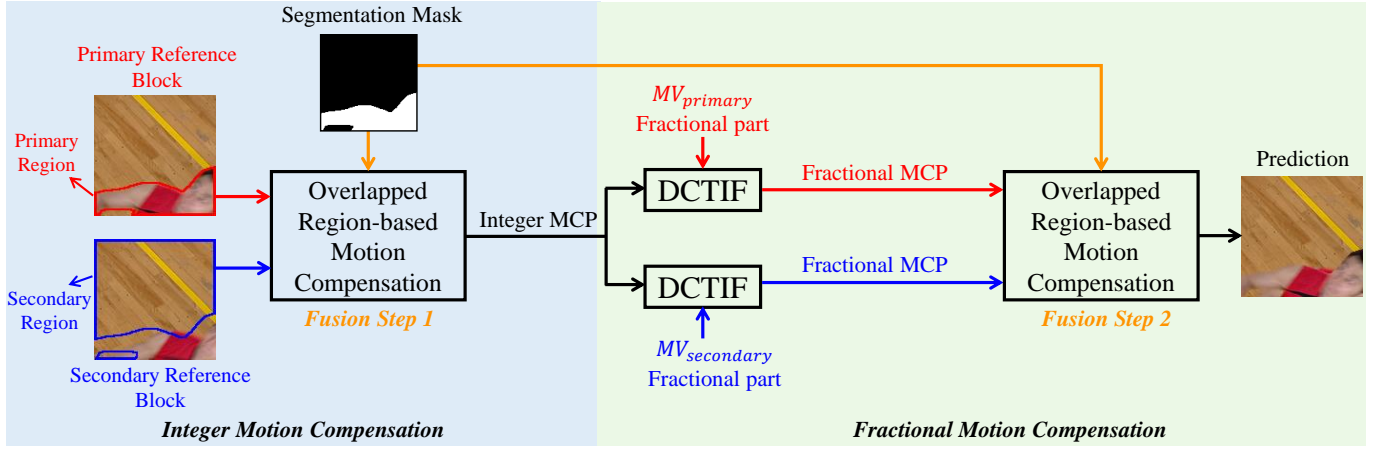
Fig. 4. Schematic representation of the segmentation-assisted motion compensation, including integer and fractional parts.
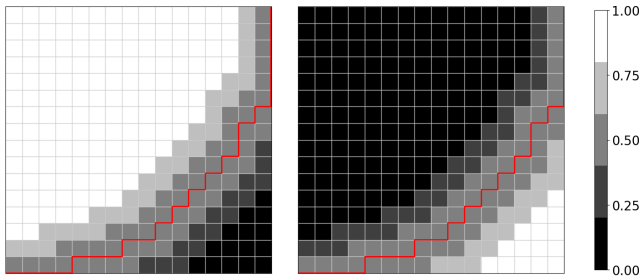

Fig. 5. An example of overlapped region-based motion compensation. The red line is the partition line of motion-inconsistent regions.

Each category corresponds to a different weight for the fusion of $P_{primary}$ and $P_{secondary}$ in every pixel.

To classify the pixels around the non-linear partition line, the segmentation mask and edge detection operator are combined to achieve accurate pixel classification. We introduce the Prewitt operator [41], [42] to classify the pixels of the segmentation mask. First, the first-order differential-based Prewitt operator is introduced to calculate the gradient between the current pixel and adjacent pixels by

$$\mathbf{G_0}(i,j) = \max\left(\mathbf{S_0} \odot \mathbf{P_0}, \mathbf{S_0} \odot \mathbf{P_0}^T\right) \qquad (2)$$

where $\mathbf{S_0}$ is a 3×3 matrix consisting of the pixel at the (i,j) position in segmentation mask and its surrounding pixels, and $\mathbf{P_0}$ is the Prewitt operator:

$$\mathbf{P_0} = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \qquad (3)$$

Then, according to the principle of the operator, the operator is expanded to 5×5 and calculate the gradient of the current pixel and the pixel at a distance of one pixel ($\mathbf{G_1}(i,j)$).

Finally, the category $\mathbf{C}$ is derived from the calculation results of $\mathbf{G_0}$ and $\mathbf{G_1}$, by

$$\mathbf{C}_{i,j} = \begin{cases} 1 & \text{if } |\mathbf{G_0}(i,j)| > 0 \\ 2 & \text{if } |\mathbf{G_0}(i,j)| = 0 \ \& \ |\mathbf{G_1}(i,j)| > 0 \\ 0 & \text{otherwise} \end{cases} \qquad (4)$$

With the pixel category $\mathbf{C}$ obtained, we allocate different fusion weights for different pixel categories. Therefore, the final prediction $\mathbf{P_{final}}$ can be generated according to the cat-

egory and the value of the segmentation mask, using different weight values to fuse $\mathbf{P_{primary}}$ and $\mathbf{P_{secondary}}$:

$$\mathbf{P_{final}}(i,j) = \begin{cases} w_0\mathbf{P_{primary}}(i,j) + w_1\mathbf{P_{secondary}}(i,j) & \text{if } \mathbf{M}_{i,j} = 1 \\ w_1\mathbf{P_{primary}}(i,j) + w_0\mathbf{P_{secondary}}(i,j) & \text{if } \mathbf{M}_{i,j} = 0 \end{cases} \qquad (5)$$

where the values of w_0 and w_1 are set by different categories:

$$\begin{cases} w_0 = \frac{1}{2}, w_1 = \frac{1}{2} & \text{if } \mathbf{C}_{i,j} = 1 \\ w_0 = \frac{3}{4}, w_1 = \frac{1}{4} & \text{if } \mathbf{C}_{i,j} = 2 \\ w_0 = 1, w_1 = 0 & \text{if } \mathbf{C}_{i,j} = 0 \end{cases} \qquad (6)$$

B. Segmentation-Assisted Fractional-Pixel Motion Compensation

In regular block-based MC, due to the inherent spatial discretization of digital video, block translation may not happen to be aligned with pixels. Thus, retrieving a block in the reference frames may not predict the coding block well enough. In VVC, discrete cosine transform-based interpolation filter (DCTIF) [43], [44] is utilized to generate the fractional-pixel MC prediction (fractional MCP) from the integer-pixel MCP and its adjacent reference pixels. In our case, due to the different motions of sub-regions within a block, the block-level fractional-pixel MC (FMC) method is mismatched for region-level FMC. Fractional-pixel prediction is needed to generate finely for each sub-region. Therefore, considering the combination of sub-regional FMC with the principle of DCTIF, we propose a two-step prediction fusion-based MC process (TWO-STEP), including integer and fractional parts. The key lies in that the integer-pixel prediction fusion is performed in IMC to assist the DCTIF of subsequent FMC. The process of fractional-pixel SA-MC is shown in Fig. 4.

First, for the integer-pixel prediction fusion, the integer-pixel MCP is generated from each sub-region's integer-pixel prediction in integer-pixel SA-MC (ORMC), and provides the precise adjacent reference pixels for the interpolation of edge-adjacent fractional pixels in the DCTIF of subsequent FMC. Second, for different sub-region's FMC, the integer-pixel MCP and the fractional part of each sub-region's MV are fed into the DCTIF as the input to generate the fractional-pixel prediction of each sub-region. Finally, with the two fractional-pixel prediction blocks obtained, the fractional-pixel prediction fusion is performed by the same strategy (ORMC) to extract
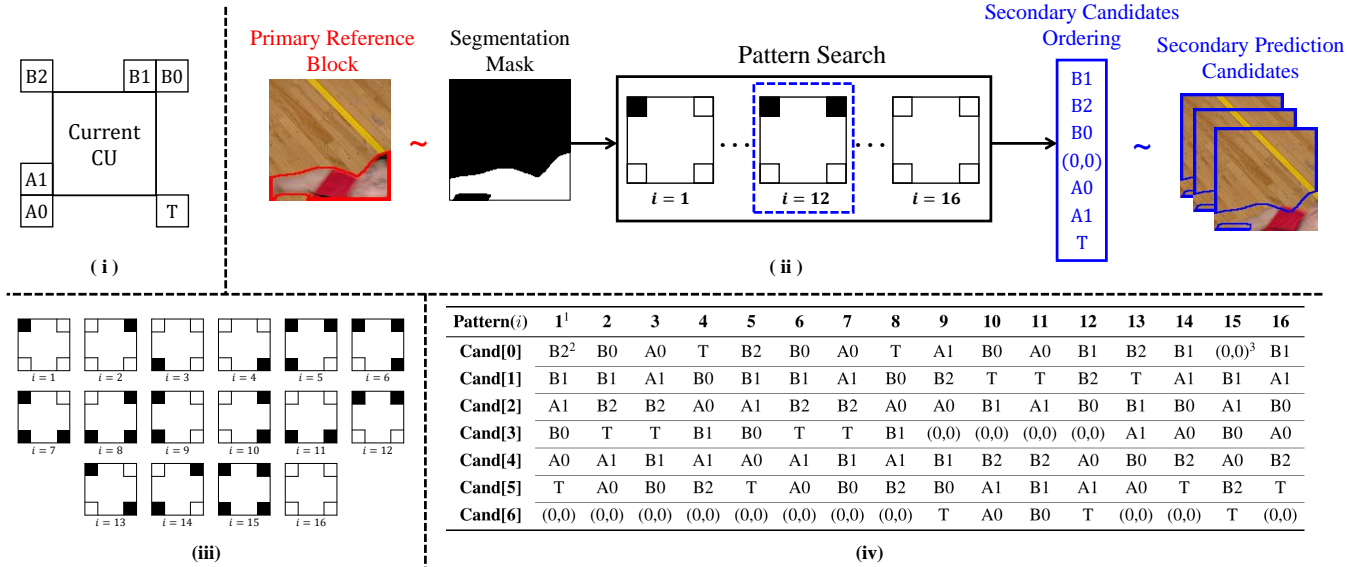
Fig. 6. (i) Position of spatial and temporal merge candidates. (ii) An example of the process of SA-MVC for a coding block. For the pattern search module, the white/black blocks in the corner represent the primary/secondary region; i indicates the pattern ID. (iii) All patterns of motion-inconsistent regions; (iv) Candidate list derivation for secondary region; Top right note 1 indicates the pattern 1 as marked in (iii); Top right note 2 indicates the neighbor CUs as marked in (i). Top right note 3 indicates zero motion vector (0, 0).

Pattern(i)	1[1]	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Cand[0]	B2[2]	B0	A0	T	B2	B0	A0	T	A1	B0	A0	B1	B2	B1	(0,0)[3]	B1
Cand[1]	B1	B1	A1	B0	B1	B1	A1	B0	B2	T	T	B2	T	A1	B1	A1
Cand[2]	A1	B2	B2	A0	A1	B2	B2	A0	A0	B1	A1	B0	B1	B0	A1	B0
Cand[3]	B0	T	T	B1	B0	T	T	B1	(0,0)	(0,0)	(0,0)	(0,0)	A1	A0	B0	A0
Cand[4]	A0	A1	B1	A1	A0	A1	B1	A1	B1	B2	B2	A0	B0	B2	A0	B2
Cand[5]	T	A0	B0	B2	T	A0	B0	B2	B0	A1	B1	A1	A0	T	B2	T
Cand[6]	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	T	A0	B0	T	(0,0)	(0,0)	T	(0,0)

and fuse the fractional MCP of different sub-regions in a block to generate the final fractional-pixel prediction.

VI. SEGMENTATION-ASSISTED MOTION VECTOR CODING

In regular inter prediction, block-based inter prediction is widely adopted, and only one MV is used to describe the motion in a coding block. It assumes that the motions within a block tend to be uniform. In regard to complex motion fields, it is difficult to follow the block-based inter mode. When the partition becomes more complicated into more blocks to represent the accurate motion field, the partition and syntax signals will increase the bits consumption. Meanwhile, the precise block partition can only roughly align with natural objects of arbitrary shape, and the real motion field is hard to capture. Thus, flexible partitioning and efficient motion representation are the keys to improving coding performance. Therefore, we propose the segmentation-assisted motion vector coding (SA-MVC) method to use the segmentation mask (object-aware partition information) to guide more accurate motion analysis for fine-grained region level, and further consider different qualities of the potential MV candidates to code the MVs of the different motion-inconsistent regions efficiently. The process of SA-MVC is shown in Fig. 6.

In SAIP, the motion of each coding block is divided into two parts (primary motion and secondary motion) to derive the motion data, which corresponds to two MVs ($MV_{primary}$, $MV_{secondary}$). One is coded by the index of the primary candidate list (length: 71), the list includes regular merge candidates (Merge) and merge with motion vector difference (MMVD) candidates [5]. The other one is coded by the index of the secondary candidate list (length: 7), the list is established with the assistance of segmentation mask.

In the coding process, the primary candidate list is established at first for the coding block, and each candidate can provide a MV ($MV_{primary}$) for the current coding block's rough motion representation. Due to multiple motions in the coding block, $MV_{primary}$ can only consider describing a

part of the motion. To further describe the multiple motions, the partition of motion area is needed. By the guidance of $MV_{primary}$, the segmentation mask of the primary reference block can be translated to the current coding block as the partition of the motion area. From the mask, the coding block can be partitioned into two motion-inconsistent regions (primary region, secondary region). For the motion of the primary region, it can be represented by $MV_{primary}$. In order to efficiently represent the motion of other regions, we propose a segmentation-assisted candidate derivation (SA-CD) strategy for the MV candidate derivation of other parts.

The construction of the regular merge candidate list follows the fixed predefined order to traverse the potential candidates (B1→A1→B0→A0→B2→T→...) to fill the candidate list. Figure 6 .(i) shows the position of spatial and temporal candidates for the current coding unit (CU), which is used to copy the motion information of the neighboring block for reference. In SA-CD, the segmentation mask (object-aware partition) is utilized as the criterion to perform motion analysis for coding block and further adaptively select and rank the high-probability candidates to construct the candidate list, thereby considering different qualities of the potential MV candidates. The example is shown in Fig. 6 .(ii). First, the distribution pattern of the secondary region in the partition of the coding block's motion-inconsistent regions is searched. The nearest four-pixel values near the corner are taken as the classification rule. In that case, the four pixels in the upper left and upper right corner of the segmentation mask are all zero (black). It is considered that these corners of the segmentation mask are in the secondary region. Similarly, the four corners can be in the secondary or the primary region, so the distribution of the motion-inconsistent regions can be classified into 16 patterns, all patterns are shown in Fig. 6 .(iii). Different patterns of distribution of motion-inconsistent regions correspond to different candidate lists. The secondary candidate list is also derived from the candidates in Fig. 6 .(i). The construction order is determined by
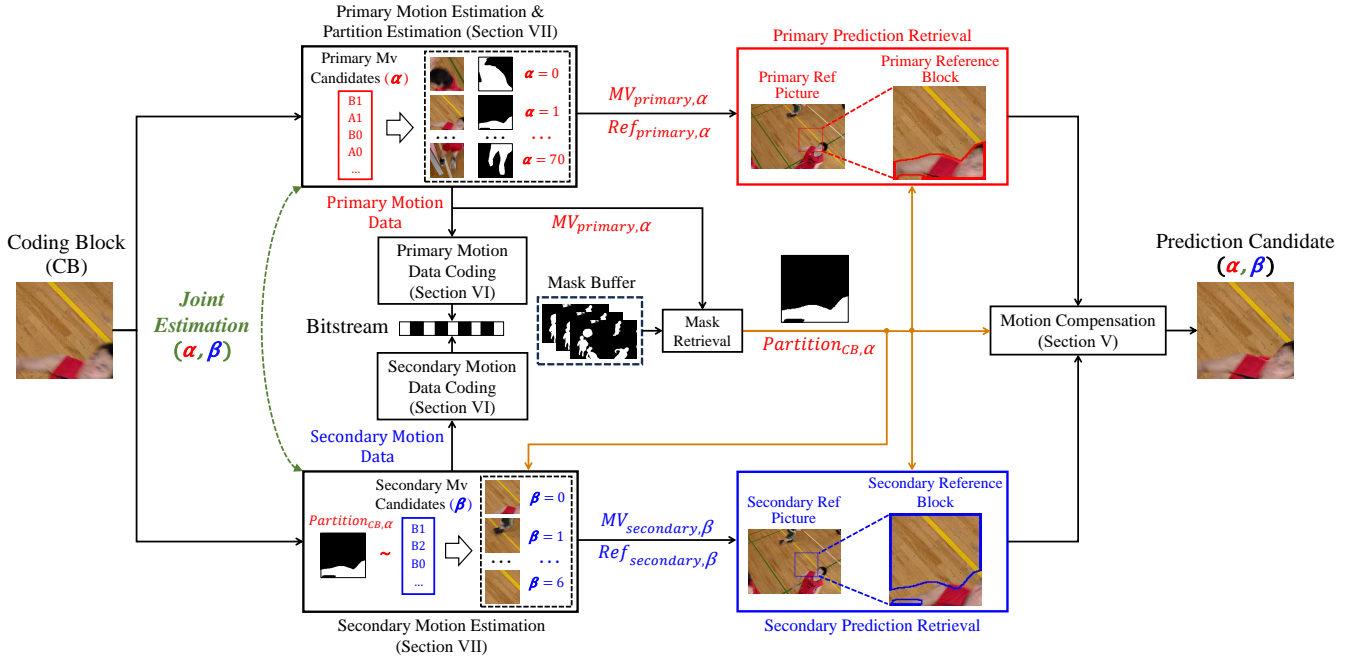
Fig. 7. Illustration of the proposed object segmentation-assisted inter prediction (SAIP) framework with segmentation-assisted rate-distortion optimization (SA-RDO) process (uni-directional case) from the perspective of encoding. Boxes represent the sub-modules of SAIP. Arrows indicate the data flow direction.

the distribution pattern of secondary region, which assumes that the motion information of neighboring blocks closer to the secondary region is more likely to match the motion information of secondary region. Therefore, according to the distribution pattern (blue dotted box) of Fig. 6 .(ii), the order can be determined as B1→B2→B0→(0,0)→A0→A1→T. The candidate list of all distribution patterns is also shown in Fig. 6 .(iv).

The motion information of the coding block also affects the coding of the subsequent coding block's ME/MC. In VVC, the motion information is stored in the cache with 4×4 unit. The stored data is used for the MV prediction of the subsequent coding blocks. To assist the MVC of the subsequent coding blocks, the motion information storage of SAIP is adaptively stored based on the partition of the motion area. The value of the segmentation mask determines the storage type of motion information (primary or secondary) for each 4×4 unit.

VII. SEGMENTATION-ASSISTED RATE-DISTORTION OPTIMIZATION FOR MOTION ESTIMATION AND PARTITION ESTIMATION

For the previous inter prediction methods of VVC, the rate-distortion optimization (RDO) for block-level partition and ME are independent of each other. In our proposed method, due to the region-level partition is introduced to assist the fine-grained inter prediction, ME for different motion-inconsistent regions and region-level partition estimation for coding block are all considered in the RDO process.

For the ME of different motion-inconsistent regions, the accuracy of motion representation depends on whether the partition (segmentation mask) of the motion area can approximate the actual motion field in the coding block. In the previous related work (GEO [11], PCS-16 [14], TIP-19 [13]), template-based or hypothesis-based partition derivation methods are utilized, the fixed search range of partition varieties and

individual RDO process (for ME and partition search) limit the partition search space of coding block and the accuracy of MVs, thereby further impacting on the prediction quality. To solve these limits, segmentation-assisted rate-distortion optimization (SA-RDO) is applied to the proposed SAIP framework. In Fig. 7, the object segmentation-assisted inter prediction framework with proposed SA-RDO (uni-directional case) is shown from the perspective of encoding. The motion candidates of $MV_{primary}$ imply various partition candidates are used to suppose the motion area partition of current coding block (partition information does not need to be explicitly signaled). And the partition estimation and the ME of primary motion and secondary motion are joint to search the best combination in the whole RDO process. Here we detail the SA-RDO from the perspective of uni-directional and bi-directional prediction.

A. Rate-Distortion Optimization for Uni-directional Prediction

In codec, the SAIP mode is added as a regular prediction mode in the mode decision list for each coding unit. For a coding block (CB) to try the uni-directional SAIP mode, first, the uni-directional primary candidate list is established. To provide the various partition candidates for accurate partition estimation, the primary candidate list is designed to be longer than the secondary candidate list. Each primary candidate contains $MV_{primary,\alpha}$, $Ref_{primary,\alpha}$ and $Partition_{CB,\alpha}$ with index $\alpha \in \{0...70\}$. Meanwhile, each primary candidate indicates a different reference block, and the different segmentation mask (representing the different regions by 1/0-value) can be translated to suppose the different motion area partition of current coding block ($Partition_{CB,\alpha}$). For the motion representation of each $MV_{primary,\alpha}$ in the coding block, it is hard to determine whether the $MV_{primary,\alpha}$ represents the 1-value region or the 0-value region's motion. Therefore, the reverse index $j \in \{0, 1\}$ is added to identify the distribution

of primary region in the segmentation mask. When the reverse index is 0, the 1-value indicates the primary region and the 0-value indicates the other parts. When the reverse index is 1, the opposite is true. Once the partition and primary region are fixed, the other region corresponds to the secondary region. Second, the uni-directional secondary candidate list is constructed according to SA-CD and Partition$_{CB,\alpha}$, the MV$_{secondary,\beta}$ and Ref$_{secondary,\beta}$ with index $\beta \in \{0 \ldots 6\}$ can be selected to represent the motion of other parts in the coding block.

With the uni-directional primary and secondary candidates constructed, the best combination of these candidates is determined by two stages.

1) Stage 1: The combined uni-directional primary and secondary candidates are used to conduct the SA-MC to generate the prediction of the coding block. The sum of the absolute transformed differences (SATD) of luma between the prediction and the original signal is computed as SATD$_{\alpha,j,\beta}$. The rate-distortion (RD) cost $J_{\alpha,j,\beta}$ for α, j and β can be sorted by

$$J_{\alpha,j,\beta} = \text{SATD}_{\alpha,j,\beta} + \lambda(R_\alpha + R_j + R_\beta), \quad (7)$$

where R_α, R_j and R_β denote the estimated rates for the primary candidate index, reverse index, and secondary candidate index, respectively.

2) Stage 2: From stage 1, the best four combined candidates further apply the residual transform coding and CABAC-based rate estimation to calculate the accurate rate cost $R_{\alpha,j,\beta}$. The distortion over three components between these candidates and the original signal is measured by the sum of squared differences (SSD) as SSD$_{\alpha,j,\beta}$. Finally, the optimal α, j, and β can be selected by

$$J_{\alpha,j,\beta} = \text{SSD}_{\alpha,j,\beta} + \lambda R_{\alpha,j,\beta}. \quad (8)$$

For the uni-directional prediction process of each coding block, 71 (the length of primary candidate list) $\times$ 7 (the length of secondary candidate list) $\times$ 2 (reverse test) = 994 times are calculated and compared in Stage 1 of SA-RDO, and 4 (the length of combined candidate list) $\times$ 2 (residual coding, use or skip) = 8 times are calculated and compared in Stage 2 of SA-RDO, 1002 times maximum calculation in total.

B. Rate-Distortion Optimization for Bi-directional Prediction

In the bi-directional case of SA-RDO, each bi-directional primary candidate contains MV$_{primary,\alpha,f}$, Ref$_{primary,\alpha,f}$, Partition$_{CB,\alpha,f}$, and MV$_{primary,\alpha,b}$, Ref$_{primary,\alpha,b}$, Partition$_{CB,\alpha,b}$ with index $\alpha \in \{0...70\}$. The f indicates the forward reference direction, and b indicates the backward reference direction. For different reference directions, the motion information of the corresponding reference direction of primary candidate indicates a reference block (retrieved by MV$_{primary,\alpha,f}$/Ref$_{primary,\alpha,f}$ or MV$_{primary,\alpha,b}$/Ref$_{primary,\alpha,b}$), and a segmentation mask (Partition$_{CB,\alpha,f}$ or Partition$_{CB,\alpha,b}$) can be translated to suppose the different motion area partition of current coding block in the corresponding reference direction. With the assistance of partition candidate (Partition$_{CB,\alpha,f}$ and
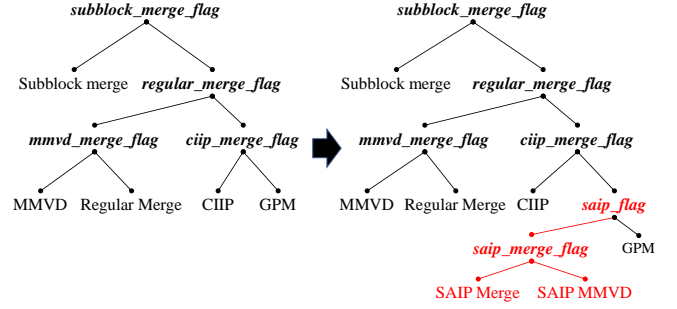

Fig. 8. Entropy coding syntax tree; the left is the original, the right is after adding the SAIP; signaled syntax elements are bold; prediction modes are shown as leaf nodes.

TABLE I
SAIP SYNTAX ELEMENTS

Syntax	Descriptor
merge_data() {	
. . .	
saip_flag	ae(v)
if(saip_flag){	
saip_merge_flag	ae(v)
if(saip_merge_flag){	
saip_merge_idx	ae(v)
}	
else{	
saip_mmvd_cand_idx	ae(v)
saip_mmvd_distance_idx	ae(v)
saip_mmvd_direction_idx	ae(v)
}	
saip_back_idx	ae(v)
saip_reverse_idx	ae(v)
}	
. . .	
}	

Partition$_{CB,\alpha,b}$) in the corresponding reference direction, the subsequent prediction process of each reference direction is the same as the uni-directional case of SA-RDO, and each bi-directional prediction candidate is obtained by the weighted average of different reference direction's prediction.

C. Complexity Optimization

To reduce the complexity of SA-RDO, we optimize the encoder algorithm in two aspects, including early termination and parameter pre-calculation. *(1) Early Termination:* The early termination is integrated into the two stages of SA-RDO to reduce the more attempts for the decision of different region's motion data. In *Stage 1*, the minimum cost of each combined candidate is recorded for subsequent candidate's early termination. For each primary candidate, after the rank of SA-CD, the highest-probability (first-order) secondary candidate with the primary candidate is tested and compared with the previous minimum cost. If higher, the attempt of subsequent secondary candidates is terminated. In *Stage 2*, for the best four combined candidates of *Stage 1*, if the cost of the first combined candidate is less than $0.9\times$ cost of the third combined candidate, the attempt of the last two combined candidates is terminated in *Stage 2*. *(2) Parameter Pre-Calculation:* To prevent the repeated calculation of weighted coefficient in SA-MC, after the reconstructed frame's segmentation mask is generated, the coefficient map is calculated in advance for subsequent frames.

TABLE II
BD-RATE RESULTS OF OUR PROPOSED SAIP METHOD COMPARED TO VTM-12.0 ON CTC TEST SEQUENCES

Class	Sequence Name	Low-delay P (%)				Low-delay B (%)				Random Access (%)			
		Y	U	V	Ratio	Y	U	V	Ratio	Y	U	V	Ratio
ClassA1 (3840x2160)	*Tango2*	-	-	-	-	-	-	-	-	-0.21%	0.21%	-0.11%	1.19%
	FoodMarket4	-	-	-	-	-	-	-	-	-0.25%	-0.25%	-0.20%	1.63%
	Campfire	-	-	-	-	-	-	-	-	-0.16%	0.02%	-0.12%	1.52%
	Average	-	-	-	-	-	-	-	-	**-0.20%**	**-0.01%**	**-0.14%**	**1.45%**
ClassA2 (3840x2160)	*CatRobot1*	-	-	-	-	-	-	-	-	-0.69%	-0.01%	-0.37%	2.12%
	DaylightRoad2	-	-	-	-	-	-	-	-	-0.21%	-0.37%	-0.38%	1.22%
	ParkRunning3	-	-	-	-	-	-	-	-	-0.06%	-0.01%	0.03%	0.72%
	Average	-	-	-	-	-	-	-	-	**-0.32%**	**-0.13%**	**-0.24%**	**1.35%**
ClassB (1920x1080)	*MarketPlace*	-0.43%	0.31%	-0.62%	2.89%	-0.46%	-0.78%	-0.99%	2.78%	-0.28%	-0.13%	-0.07%	1.36%
	RitualDance	-0.23%	-0.35%	-0.48%	2.15%	-0.28%	0.42%	0.17%	2.27%	-0.18%	0.27%	-0.12%	1.21%
	Cactus	-0.28%	-0.47%	-0.38%	2.34%	-0.28%	-0.24%	-0.19%	2.37%	-0.30%	0.33%	-0.40%	1.58%
	BasketballDrive	-0.29%	-0.43%	-0.66%	2.52%	-0.25%	-0.28%	-0.33%	1.69%	-0.20%	0.75%	-0.21%	1.45%
	BQTerrace	-0.03%	-0.09%	-0.11%	0.73%	-0.19%	0.47%	0.49%	1.68%	-0.34%	-0.19%	-0.40%	1.11%
	Average	**-0.25%**	**-0.21%**	**-0.45%**	**2.13%**	**-0.30%**	**-0.08%**	**-0.17%**	**2.16%**	**-0.26%**	**0.21%**	**-0.24%**	**1.34%**
ClassC (832x480)	*BasketballDrill*	-1.83%	-1.03%	-1.83%	10.73%	<u>-1.14%</u>	-0.74%	-1.00%	7.59%	<u>-0.79%</u>	-0.08%	-0.41%	4.45%
	BQMall	<u>-1.98%</u>	-2.84%	-2.41%	10.39%	-0.84%	-1.07%	-0.63%	6.54%	-0.66%	-1.08%	-0.70%	4.85%
	PartyScene	-0.27%	-0.61%	-1.03%	2.92%	-0.17%	-0.36%	-0.16%	2.67%	-0.17%	-0.27%	0.03%	2.29%
	RaceHorsesC	-0.95%	-1.23%	-0.39%	5.75%	-0.23%	-0.13%	0.30%	3.55%	-0.37%	-0.40%	-0.06%	2.72%
	Average	**-1.26%**	**-1.43%**	**-1.41%**	**7.45%**	**-0.60%**	**-0.58%**	**-0.37%**	**5.09%**	**-0.50%**	**-0.46%**	**-0.29%**	**3.58%**
ClassE (1280x720)	*FourPeople*	-0.93%	-0.64%	-0.17%	3.56%	-0.46%	0.29%	-0.36%	2.47%	-0.41%	-0.35%	-0.21%	1.37%
	Johnny	-1.89%	-1.29%	-1.30%	5.58%	-1.12%	-2.02%	0.46%	3.54%	-0.67%	-0.25%	-0.34%	2.30%
	KristenAndSara	-0.68%	-1.63%	0.02%	3.35%	-0.38%	-0.80%	0.66%	2.19%	-0.48%	-0.77%	-0.18%	1.36%
	Average	**-1.17%**	**-1.19%**	**-0.48%**	**4.16%**	**-0.66%**	**-0.84%**	**0.25%**	**2.73%**	**-0.52%**	**-0.46%**	**-0.24%**	**1.68%**
Overall		**-0.82%**	**-0.86%**	**-0.78%**	**4.58%**	**-0.49%**	**-0.45%**	**-0.13%**	**3.33%**	**-0.37%**	**-0.14%**	**-0.23%**	**1.88%**
ClassD (416x240)	*BasketballPass*	-0.61%	-0.38%	-1.17%	8.58%	-0.48%	-0.97%	-0.74%	5.63%	-0.44%	0.55%	-0.21%	2.78%
	BQSquare	-0.78%	-2.28%	3.65%	5.82%	-0.42%	-1.73%	1.76%	4.13%	-0.53%	-0.60%	-0.37%	2.68%
	BlowingBubbles	-1.02%	-1.17%	-1.98%	5.85%	-0.63%	-0.16%	0.39%	4.41%	-0.50%	-0.81%	-1.06%	3.40%
	RaceHorses	-0.83%	-0.21%	-0.90%	6.28%	-0.10%	0.88%	0.15%	3.68%	-0.34%	0.94%	0.49%	3.03%
	Average	**-0.81%**	**-1.01%**	**-0.10%**	**6.63%**	**-0.41%**	**-0.49%**	**0.39%**	**4.46%**	**-0.45%**	**0.02%**	**-0.29%**	**2.97%**

VIII. SYNTAX DESIGN

Syntax elements of SAIP are coded into the bitstream using entropy coding. In the high-level syntax (HLS), two SAIP enabling flags (SAIP_Merge, SAIP_MMVD) and the maximum candidate number of the secondary candidate list are coded in the sequence parameter set. The enabling flags are coded with fixed-length code, and the maximum candidate number is coded with the 0-th order exponential-Golomb code.

At the coding unit level, the SAIP syntax elements are signaled inside the merge data syntax. Figure 8 shows the changes in the syntax tree of merge data before and after adding SAIP syntax elements. To minimize the impact of the newly increased syntax elements on coding the syntax of other modes, the original syntax tree of merge data has been redesigned. In the new syntax elements tree, saip_flag is signaled to indicate whether it is in SAIP mode or GPM (GEO) mode, and saip_merge_flag is signaled to indicate whether it is in SAIP-Merge mode or SAIP-MMVD mode.

Table I is the syntax elements table for SAIP. It contains four parts, including the syntax elements of the primary region (Merge: *saip_merge_idx*, MMVD: *saip_mmvd_cand_idx*, *saip_mmvd_distance_idx* and *saip_mmvd_direction_idx*), secondary region (*saip_back_idx*), mode flag (*saip_flag* and *saip_merge_flag*), and the reverse index (*saip_reverse_idx*). These elements are coded by using the context-based adaptive binary arithmetic coding (CABAC) engine. To code the newly increased syntax elements efficiently, the mode flag, reverse index, and the syntax elements of the secondary region are coded by using a new context model.

IX. EXPERIMENTS

A. Experimental Settings

1) Learned Segmentation Inference: For image instance segmentation, following [31], we use the pre-trained SCNet model and infer the model on the MMDetection platform [45]. During inference, object proposals are progressively refined by box branches at different stages. The final classification score is obtained from the scores of multiple classifiers. To exclude object classification error, only objects whose classification score exceeds the threshold $T \in (0, 1)$ can be segmented. In our experiments, T is set to 0.3, 0.6, and 0.9. For video object segmentation, following [37], the STCN adopts the inference code and pre-trained model [1], and follows the space-time memory networks of the STM [36], which stores the previous frames and masks in the memory bank. When inferring the mask of the current frame, five frames are selected from the memory bank and input with the current frame into the STCN.

2) Evaluation Configurations: To evaluate the performance of SAIP in VVC, SAIP is implemented into the VVC reference

---

[1] https://github.com/hkchengrex/STCN

TABLE III
TIME COMPLEXITY OF OUR METHOD COMPARED TO VTM ANCHOR

Class	Low-delay P		Low-delay B		Random Access	
	EncT	DecT	EncT	DecT	EncT	DecT
A	-	-	-	-	183%	116%
B	179%	111%	171%	112%	180%	112%
C	159%	109%	147%	113%	154%	115%
E	178%	113%	148%	120%	179%	118%
Overall	**172%**	**111%**	**155%**	**115%**	**174%**	**115%**
D	135%	108%	123%	119%	116%	117%

TABLE IV
COMPARISONS WITH SOME ENHANCED GEOMETRIC-BASED
PARTITIONING METHODS UNDER LOW-DELAY B AND RANDOM ACCESS
CONFIGURATIONS. NOTE THAT DIFFERENT METHODS ARE COMPARED TO
DIFFERENT ANCHORS (THE ANCHOR IS SHOWN IN PARENTHESES, SUCH
AS VTM-8.0 FOR TIP-21, VTM-12.0 FOR SAIP).

Class	Low-delay B			Random Access		
	TIP-21[1]	VTM-12.0	SAIP[2]	TIP-21[1]	VTM-12.0	SAIP[2]
	(VTM-8.0)	(VTM-8.0)	(VTM-12.0)	(VTM-8.0)	(VTM-8.0)	(VTM-12.0)
A1	-	-	-	-0.20%	-5.69%	-0.20%
A2	-	-	-	-0.21%	-12.08%	-0.32%
B	-0.31%	-1.17%	-0.30%	-0.19%	-6.96%	-0.26%
C	-0.61%	-1.41%	-0.60%	-0.32%	-3.22%	-0.50%
E	-0.88%	-1.29%	-0.66%	-	-8.35%	-0.52%
Overall	**-0.55%**	**-1.28%**	**-0.49%**	**-0.23%**	**-7.00%**	**-0.37%**
D	-	-1.02%	-0.41%	-	-0.83%	-0.45%

[1] These results are cited from [20]. Only partial sequences were reported.
[2] The performance of SAIP on top of VTM-12.0.

software VTM-12.0 [2]. Three main profile configurations (Low-delay P (LDP), Low-delay B (LDB), Random Access (RA)) and test sequences specified in [46] are used as the test conditions. Following the common test conditions (CTC) [46], four quantization parameters are tested: 22, 27, 32 and 37. For different quantization parameters, different segmentation configurations of the SCNet and STCN are chosen. For the measure of coding performance, the Bjontegaard Delta bit-rate (BD-rate) [47] is used as the objective metric to evaluate it. To further measure the characteristic of the proposed method on different sequences, the usage ratio ($Ratio$) is used to validate the efficiency of mode selection, which is calculated by, $Ratio = N_{test}/N_{total}$, where N_{test} indicates the number of coding units coded by the proposed method in P/B-frames, and N_{total} indicates the total number of coding units in P/B-frames of the test sequences.

B. Performance

In this subsection, we show the overall performance of the proposed object segmentation-assisted inter prediction. First, we show the coding performance and encoding/decoding complexity of the proposed SAIP implemented into the VVC reference software VTM-12.0 under CTC [46]. Second, we compare SAIP with previous related methods to verify its superior performance. Finally, we test SAIP on the test sequence of specific scenes to further demonstrate its effectiveness.

1) Overall Performance Under Common Test Conditions: The R-D performance of the entire segmentation-assisted inter prediction framework on the VVC common test sequences is illustrated in Table II. Y, U, and V represent the R-D performance gain of the three channels of YUV. We can see that our proposed SAIP can achieve on average, 0.82%, 0.49% and 0.37% (marked by font bold), and achieve up to 1.98%, 1.14% and 0.79% (marked by underline) BD-rate reduction (Y component) with high usage ratio on VTM-12.0 for all sequences under the LDP, LDB, and RA configurations. The experimental results show that the proposed framework performs better for sequences with more moving objects and complex motion fields, such as *BasketballDrill*, *BQMall*, *CatRobot*, and *Marketplace*.

In VVC reference software (VTM-12.0), the technology of inter prediction most relevant to SAIP is the geometric partitioning mode (GEO) [11]. Both ours and GEO aim to

[2]https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM

achieve a flexible description of the motion field and match the appropriate prediction for different regions in a block. In Table II, under the LDB and RA configurations, GEO is enabled in the anchor by default. In the comparison with anchor, SAIP achieves good performance with GEO enabled. It can be concluded that SAIP can efficiently deal with some complex motion scenarios that GEO is difficult to address.

2) Complexity: The encoding/decoding time complexity of our proposed method is illustrated in Table III. EncT and DecT represent the encoding/decoding time with the inference time of segmentation network, the network is inferred in parallel with the syntax parsing. The codec time is tested on CPU, and the model is Intel Core i7-11700 @2.50GHz. The inference time of network is tested on GPU, the models are aligned with the [31], [37], and the computational complexity is 119 kMAC/pixel.

Compared with the anchor, under LDP, LDB and RA, for the decoding time, our proposed method does not obviously increase the time complexity in the decoding process. The slight increase in DecT is mainly due to the inference time of the segmentation network. For the encoding time, the increase in EncT is mainly due to the SA-RDO for the partition estimation of coding block and ME of primary and secondary regions to search accurate motion area partition and the MVs of each region.

3) Comparisons with State-of-the-Art Methods: Table V and VI show the experimental data comparing our proposed SAIP with the previous related work on segmentation-based partition methods [13] and [14], under LDP, LDB and RA configurations ([13] did not report RA configuration, [14] did not report LDP configuration). These related methods use different anchors. Specifically, [13] and [14] use HM-14.0 and VTM-1.0 as the anchor, respectively. We directly cite the results reported in [13] and [14], since our reimplementation may not work as well as the authors'. HM-14.0 [48] and VTM-1.0 [49] do not contain some new technical tools in the latest version of VVC, which have a large performance gap with VTM-12.0. To make a fair comparison with related work, we test SAIP in a simplified VTM-12.0 with some coding tools turned off, and compare it with [13] [14] on HEVC and VVC test sequences. Note that the turned-off coding tools in

TABLE V
COMPARISONS WITH THE OTHER STATE-OF-THE-ART METHODS UNDER LOW-DELAY B CONFIGURATION. NOTE THAT DIFFERENT METHODS ARE COMPARED TO DIFFERENT ANCHORS (THE ANCHOR IS SHOWN IN PARENTHESES, SUCH AS HM-14.0 FOR PCS-16, VTM-1.0 FOR TIP-19).

Sequence	PCS-16[1] (HM-14.0)	TIP-19[2] (VTM-1.0)			Simplified VTM-12.0[3] (HM-14.0)			Simplified VTM-12.0[3] (VTM-1.0)			SAIP[4] (Simplified VTM-12.0[3])		
Name	Y	Y	U	V	Y	U	V	Y	U	V	Y	U	V
MarketPlace	-	-	-	-	-16.81%	-12.53%	-20.41%	-4.86%	20.24%	22.04%	-2.38%	-2.09%	-2.38%
RitualDance	-	-	-	-	-25.05%	-20.87%	-20.66%	-8.90%	17.14%	16.08%	-2.15%	-1.98%	-1.91%
Cactus	-	-1.00%	-0.90%	-1.00%	-20.36%	-6.45%	-6.35%	-9.88%	22.29%	22.37%	-0.80%	-0.77%	-0.86%
BasketballDrive	-0.08%	-0.70%	-0.80%	-1.10%	-24.02%	-21.09%	-21.25%	-9.47%	27.71%	23.75%	-2.20%	-3.00%	-2.88%
BQTerrace	-	-1.60%	-2.00%	-1.70%	-16.10%	-15.34%	-28.19%	-8.08%	18.61%	11.56%	-1.48%	-1.01%	-0.94%
Kimono	-0.14%	-0.90%	-1.00%	-1.40%	-15.01%	4.40%	0.68%	-6.15%	20.14%	20.14%	-1.17%	-1.00%	-0.87%
ParkScene	-	-0.60%	-0.40%	-0.70%	-16.60%	-3.55%	-4.14%	-6.89%	20.26%	22.02%	-0.23%	-0.63%	-0.55%
BasketballDrill	-	-1.30%	-1.50%	-1.30%	-29.14%	-15.11%	-14.63%	-19.76%	9.37%	10.45%	-4.34%	-4.27%	-4.91%
BQMall	-	-1.50%	-1.10%	-1.00%	-22.74%	-14.37%	-13.82%	-12.03%	14.86%	14.43%	-4.47%	-5.91%	-5.14%
PartyScene	-	-1.80%	-1.40%	-1.10%	-16.82%	0.69%	-1.07%	-9.07%	21.47%	22.33%	-1.16%	-1.10%	-1.37%
RaceHorsesC	-0.95%	-0.60%	-0.90%	-0.80%	-12.52%	10.91%	10.28%	-5.06%	30.48%	34.81%	-2.21%	-2.97%	-3.63%
BasketballPass	-	-0.80%	-0.80%	-1.00%	-17.88%	-10.36%	-6.16%	-7.69%	23.35%	24.07%	-2.41%	-3.60%	-4.02%
BQSquare	-	-1.90%	-1.60%	-2.00%	-13.55%	-10.62%	-12.38%	-8.34%	7.68%	5.87%	-1.26%	0.17%	1.14%
BlowingBubbles	-0.30%	-1.70%	-1.40%	-1.60%	-12.62%	6.62%	2.90%	-5.55%	24.08%	24.18%	-2.18%	-2.35%	-1.14%
RaceHorses	-	-1.00%	-1.20%	-1.00%	-13.47%	7.69%	7.84%	-5.10%	29.82%	29.66%	-2.11%	-2.81%	-2.30%
FourPeople	-	-0.70%	-0.60%	-1.20%	-25.13%	-28.39%	-29.28%	-10.46%	0.92%	-0.80%	-2.23%	-1.48%	-1.86%
Johnny	-	-0.90%	-1.00%	-1.20%	-25.81%	-30.38%	-34.35%	-10.25%	9.35%	6.11%	-3.80%	-3.04%	-2.73%
KristenAndSara	-	-1.10%	-1.30%	-0.90%	-23.97%	-26.32%	-28.35%	-8.89%	6.64%	5.79%	-1.53%	-1.07%	2.21%
Average	**-0.37%**	**-1.13%**	**-1.12%**	**-1.13%**	**-20.72%**	**-12.74%**	**-15.11%**	**-8.69%**	**18.02%**	**17.49%**	**-2.12%**	**-2.16%**	**-1.90%**

[1] These results are cited from [13]. Only Y-channel BD-rate of partial sequences was reported.
[2] These results are cited from [14]. Only partial sequences were reported.
[3] VTM-12.0 with several coding tools turned off (MTS, SBT, LFNST, ISP, MMVD, Affine, SbTMVP, LMChroma, DepQuant, IMV, ALF, CCALF, BCW, BcwFast, BIO, CIIP, Geo, AffineAmvr, LMCS, MRL, MIP, DMVR, SMVD, JointCbCr, PROF, ChromaTS). Note that the turned-off coding tools are all not supported in HM-14.0 [48] and VTM-1.0 [49], and do not affect the inter technology of HM-14.0 and VTM-1.0.
[4] The performance of SAIP on top of Simplified VTM-12.0.

TABLE VI
COMPARISONS WITH THE OTHER STATE-OF-THE-ART METHODS UNDER LOW-DELAY P AND RANDOM ACCESS CONFIGURATIONS. NOTE THAT DIFFERENT METHODS ARE COMPARED TO DIFFERENT ANCHORS (THE ANCHOR IS SHOWN IN PARENTHESES, SUCH AS HM-14.0 FOR PCS-16, VTM-1.0 FOR TIP-19).

Sequence	Low-delay P Configuration (%)			Random Access Configuration (%)									
	PCS-16[1] (HM-14.0)	Simplified VTM-12.0[3] (HM-14.0)	SAIP[4] (Simplified VTM-12.0[3])	TIP-19[2] (VTM-1.0)			Simplified VTM-12.0[3] (VTM-1.0)			SAIP[4] (Simplified VTM-12.0[3])		
Name	Y	Y	Y	Y	U	V	Y	U	V	Y	U	V
MarketPlace	-	-18.51%	-2.50%	-	-	-	-13.80%	0.31%	0.12%	-4.25%	-3.40%	-3.37%
RitualDance	-	-25.78%	-2.28%	-	-	-	-7.38%	21.24%	21.23%	-2.23%	-2.13%	-2.00%
Cactus	-	-21.26%	-1.24%	-0.90%	-0.70%	-0.90%	-15.97%	1.45%	4.13%	-4.39%	-4.08%	-4.00%
BasketballDrive	-0.18%	-25.95%	-1.98%	-0.90%	-0.80%	-1.10%	-13.14%	13.33%	11.14%	-3.17%	-3.29%	-2.89%
BQTerrace	-	-23.45%	-0.78%	-1.50%	-1.20%	-1.30%	-23.84%	-8.67%	-10.70%	-2.13%	-1.62%	-1.55%
Kimono	+0.04%	-19.11%	-0.82%	-1.10%	-1.30%	-0.80%	-7.64%	14.96%	13.62%	-2.23%	-2.78%	-2.32%
ParkScene	-	-16.45%	-0.57%	-1.10%	-1.20%	-0.70%	-11.18%	7.55%	9.02%	-1.71%	-1.52%	-1.21%
BasketballDrill	-	-31.04%	-3.00%	-1.40%	-2.10%	-1.80%	-12.05%	12.68%	10.29%	-2.70%	-2.99%	-3.24%
BQMall	-	-24.41%	-3.69%	-1.60%	-1.40%	-1.60%	-10.11%	13.54%	13.47%	-4.38%	-4.87%	-5.01%
PartyScene	-	-20.51%	-1.50%	-1.50%	-1.00%	-0.70%	-5.46%	17.11%	17.49%	-1.68%	-1.94%	-1.93%
RaceHorsesC	-1.05%	-14.49%	-2.60%	-0.70%	-0.50%	-0.40%	-10.85%	18.34%	19.93%	-2.58%	-4.09%	-4.42%
BasketballPass	-	-17.59%	-2.79%	-0.90%	-1.10%	-1.00%	-8.09%	18.51%	17.10%	-1.66%	-2.28%	-2.27%
BQSquare	-	-23.14%	-2.64%	-1.60%	-2.40%	-2.20%	-4.37%	5.95%	7.04%	-2.17%	-2.07%	-1.50%
BlowingBubbles	-0.17%	-14.15%	-2.27%	-1.80%	-1.90%	-2.10%	-7.52%	15.82%	17.49%	-2.43%	-2.00%	-2.66%
RaceHorses	-	-14.06%	-2.99%	-0.80%	-0.70%	-0.70%	-6.52%	20.78%	18.72%	-2.05%	-3.04%	-2.75%
FourPeople	-	-26.19%	-1.26%	-0.80%	-0.60%	-0.90%	-14.32%	-6.64%	-8.85%	-1.82%	-1.37%	-1.29%
Johnny	-	-28.65%	-3.38%	-0.80%	-0.50%	-0.20%	-16.01%	-5.92%	-10.13%	-3.41%	-2.80%	-3.03%
KristenAndSara	-	-24.97%	-1.62%	-1.00%	-0.70%	-0.90%	-14.19%	-4.63%	-5.28%	-2.07%	-1.63%	-1.56%
Tango2	-	-	-	-0.80%	-1.10%	-1.20%	-11.59%	4.54%	3.47%	-2.90%	-3.32%	-2.85%
Drums100	-	-	-	-1.40%	-1.80%	-1.60%	-17.93%	-2.46%	-1.84%	-1.78%	-1.93%	-1.23%
Campfire	-	-	-	-1.70%	-2.50%	-3.30%	-20.34%	9.60%	8.37%	-0.72%	-0.47%	-0.88%
ToddlerFountain2	-	-	-	-0.80%	-0.90%	-1.30%	-6.43%	48.16%	38.69%	-0.23%	-1.59%	-0.57%
TrafficFlow	-	-	-	-0.90%	-1.10%	-0.90%	-14.30%	2.16%	2.74%	-1.44%	-1.35%	-1.29%
DaylightRoad2	-	-	-	-1.10%	-1.00%	-1.10%	-17.23%	-8.60%	-7.49%	-3.36%	-2.47%	-2.54%
FoodMarket4	-	-	-	-	-	-	-9.80%	10.05%	11.89%	-1.83%	-1.81%	-1.21%
CatRobot	-	-	-	-	-	-	-15.13%	0.94%	1.21%	-4.24%	-5.45%	-5.28%
ParkRunning3	-	-	-	-	-	-	-17.23%	9.61%	12.50%	-3.92%	-4.07%	-4.32%
Average	**-0.34%**	**-22.14%**	**-2.10%**	**-1.14%**	**-1.20%**	**-1.22%**	**-12.31%**	**8.51%**	**7.98%**	**-2.50%**	**-2.61%**	**-2.49%**

Note: The indication of table notes [1], [2], [3], [4] is the same as in Table V.

TABLE VII
BD-RATE RESULTS OF OUR PROPOSED SAIP COMPARED TO VTM-12.0 ON SPECIFIC SCENES.

Sequence	Resolution	Low-Delay P (%)			Low-Delay B (%)			Random Access (%)		
		Y	U	V	Y	U	V	Y	U	V
Carphone	*QCIF*	-1.67%	-0.09%	2.57%	-1.05%	-1.85%	0.63%	-0.80%	-1.33%	-0.26%
Grandma	*QCIF*	-1.26%	-1.91%	-0.72%	-1.27%	-0.90%	-1.41%	-0.18%	-0.56%	-0.08%
Deadline	*CIF*	-0.92%	-0.81%	-1.89%	-0.81%	-0.44%	0.81%	-0.67%	-0.63%	-1.19%
Akiyo	*CIF*	-0.69%	0.05%	-0.50%	-0.86%	-1.46%	-1.02%	-0.01%	0.24%	0.41%
Silent	*CIF*	-2.90%	-4.03%	-2.78%	-1.37%	-1.26%	-0.62%	-1.02%	-0.45%	-0.57%
Bowing	*CIF*	-1.71%	-3.15%	-1.58%	-1.08%	-0.48%	-1.40%	-0.72%	-0.37%	-0.69%
Football	*CIF*	-0.71%	1.39%	-2.06%	-0.70%	-1.24%	0.38%	-0.63%	0.83%	-0.47%
Foreman	*CIF*	-0.85%	0.60%	-2.98%	-0.52%	-1.96%	1.11%	-0.38%	-0.38%	-0.80%
Vidyo1	*720P*	-0.75%	-0.19%	-1.06%	-0.81%	0.16%	-0.39%	-0.58%	-0.75%	-0.51%
Vidyo3	*720P*	-1.27%	-1.80%	-0.88%	-0.80%	-1.22%	-0.29%	-0.63%	-0.45%	-0.33%
Vidyo4	*720P*	-0.84%	-0.07%	0.65%	-0.55%	-0.59%	-0.99%	-0.43%	-0.37%	-0.19%
Crew	*720P*	-0.73%	1.25%	-1.20%	-0.59%	-1.48%	-0.98%	-0.52%	-1.82%	0.51%
Aspen	*1080P*	-0.51%	-0.20%	-0.37%	-0.42%	-0.53%	0.12%	-0.36%	-0.17%	0.08%
Red_kayak	*1080P*	-0.50%	-0.30%	0.86%	-0.51%	-1.86%	0.61%	-0.47%	0.20%	-1.02%
Sunflower	*1080P*	-0.60%	-0.92%	-0.86%	-0.64%	-0.72%	-0.42%	-0.49%	0.01%	-0.25%
FoodMarket2	*4K*	-0.45%	-0.63%	-0.21%	-0.47%	-1.13%	0.36%	-0.51%	-0.32%	0.24%
BoxingPractice	*4K*	-1.22%	-1.72%	-0.70%	-0.90%	-1.67%	-1.15%	-0.78%	-1.23%	-0.55%
Crosswalk	*4K*	-0.67%	0.18%	-0.85%	-0.63%	-1.86%	-0.35%	-0.31%	-0.70%	-0.71%
Narrator	*4K*	-1.31%	-1.68%	-2.15%	-1.08%	-1.66%	-1.02%	-0.34%	0.18%	0.10%
SquareAndTimelapse	*4K*	-0.65%	-0.03%	-0.96%	-0.55%	0.07%	-0.54%	-0.50%	-0.68%	-0.47%
Overall	**-**	**-1.01%**	**-0.70%**	**-0.88%**	**-0.78%**	**-1.10%**	**-0.33%**	**-0.52%**	**-0.44%**	**-0.34%**

simplified VTM-12.0 are all not supported in HM-14.0 [48] and VTM-1.0 [49]. (The number of coded frames of these methods in Table V and VI follows the number of coded frames specified in the Common Test Condition of HEVC [50] and VVC [46].) From Table V and VI, it is observed that the proposed SAIP achieves a better relative BD-rate reduction on higher version than the other methods on low versions, which demonstrates the superior performance of the proposed method.

In terms of performance comparison with previous segmentation-based partition methods [13] [14], the bitrate savings of the proposed method mainly come from the following four aspects: (1) We design a complete segmentation-assisted inter prediction framework, which sufficiently explores the assistance effect of segmentation in each sub-modules of inter prediction. In addition, the flexible and accurate motion area partition greatly improves the efficiency of inter prediction for motion-inconsistent regions. (2) For the MC of multiple motion-inconsistent regions in a block, the segmentation-assisted process can achieve higher prediction accuracy. It leverages the assistance of segmentation for the multi-stage MC. (3) For the MVC of different regions, the proposed SA-CD fully utilizes the segmentation mask to provide suitable prediction candidates efficiently and further reduces the transmission of motion data. (4) For the ME of different regions, the motion information candidates implicitly indicate partition candidates for the partition estimation of coding block, which provide various partition candidates to suppose the actual motion situation and jointly consider the partition estimation and ME in the RDO process to estimate the partition and MVs accurately.

In addition, we also compare our proposed method with some enhanced geometric-based partitioning methods [20],

under LDB and RA configurations. The comparison result is shown in Table IV. Specifically, [20] uses VTM-8.0 as the anchor, and we directly cite the results reported in [20] (did not report LDP configuration). From Table IV, it is observed that the proposed SAIP achieves a good relative BD-rate reduction on the VTM higher version than the other enhanced geometric-based partitioning methods on the VTM low version, which further verifies the efficiency of the proposed method.

4) On Specific Scenes: According to the performance of SAIP under CTC and the comparison with other methods, the proposed object-aware prediction framework is beneficial to the scene with moving objects and complex motion fields (such as the walkers of *BQMall* shown in Fig. 9 and the athletes of *BasketballDrill*), and achieves high performance on these scenes. To further verify the characteristics and advantages of SAIP, we test SAIP on some selected test sequences of specific scenes (diverse scenes, multiple moving objects, complex motions) from previous standards' CTC [1], [7] and popular datasets [51]. Table VII shows the performance of partial sequences on these specific scenes. We can see that our proposed SAIP can achieve up to 2.90%, 1.37%, 1.02%, and on average 1.01%, 0.78%, 0.52% BD-rate reduction (Y component) on VTM-12.0. The results demonstrate the SAIP can achieve accurate object-aware prediction on specific scenes with more moving objects, and further improve the flexibility of motion modeling in VVC.

C. Performance Analysis

In this subsection, we analyze the performance of SAIP through the effectiveness of core modules and the mode selection results of SAIP in detail.

1) Ablation Study: To demonstrate the contributions of two core modules in our scheme, we conduct the ablation experiments on proposed SA-MC and SA-MVC.

TABLE VIII
ABLATION STUDY ON SA-MC AND SA-MVC BASED ON VTM-12.0

Low-delay P Configuration (%)											
Class	**ONE-STEP SA-MC**			**TWO-STEP SA-MC**			**w/o SA-MVC**			**SAIP (SA-MC+SA-MVC)**		
	Y	U	V	Y	U	V	Y	U	V	Y	U	V
B	-0.05%	0.03%	-0.29%	-0.10%	-0.17%	-0.11%	-0.11%	-0.48%	-0.20%	-0.25%	-0.21%	-0.45%
C	-0.52%	-0.64%	-0.69%	-0.89%	-1.21%	-0.70%	-1.07%	-1.23%	-1.31%	-1.26%	-1.43%	-1.41%
E	-0.49%	-0.51%	-0.31%	-0.68%	-0.90%	-0.74%	-0.64%	-1.46%	-0.48%	-1.17%	-1.19%	-0.48%
Avg.	**-0.32%**	**-0.33%**	**-0.43%**	**-0.51%**	**-0.70%**	**-0.46%**	**-0.56%**	**-0.98%**	**-0.64%**	**-0.82%**	**-0.86%**	**-0.78%**
D	-0.58%	-0.52%	-0.93%	-0.83%	-0.89%	-1.28%	-0.60%	-0.64%	-0.64%	-0.81%	-1.01%	-0.10%

Low-delay B Configuration (%)											
Class	**ONE-STEP SA-MC**			**TWO-STEP SA-MC**			**w/o SA-MVC**			**SAIP (SA-MC+SA-MVC)**		
	Y	U	V	Y	U	V	Y	U	V	Y	U	V
B	0.01%	-0.08%	0.14%	-0.11%	-0.13%	0.00%	-0.12%	-0.37%	0.11%	-0.30%	-0.08%	-0.17%
C	-0.07%	0.00%	-0.29%	-0.42%	-0.79%	-0.08%	-0.32%	-0.37%	-0.75%	-0.60%	-0.58%	-0.37%
E	-0.33%	-0.12%	-0.37%	-0.43%	-0.62%	-0.20%	-0.27%	-0.45%	-0.07%	-0.66%	-0.84%	0.25%
Avg.	**-0.10%**	**-0.06%**	**-0.13%**	**-0.29%**	**-0.47%**	**-0.08%**	**-0.22%**	**-0.39%**	**-0.22%**	**-0.49%**	**-0.45%**	**-0.13%**
D	-0.37%	-0.65%	0.45%	-0.47%	0.14%	0.35%	-0.12%	0.21%	0.93%	-0.41%	-0.49%	0.39%

Random Access Configuration (%)											
Class	**ONE-STEP SA-MC**			**TWO-STEP SA-MC**			**w/o SA-MVC**			**SAIP (SA-MC+SA-MVC)**		
	Y	U	V	Y	U	V	Y	U	V	Y	U	V
A	0.80%	0.31%	0.41%	-0.06%	-0.13%	-0.13%	-0.08%	0.02%	0.02%	-0.26%	-0.07%	-0.19%
B	0.76%	0.55%	0.65%	-0.05%	0.07%	-0.10%	-0.09%	-0.04%	-0.04%	-0.26%	0.21%	-0.24%
C	0.55%	0.34%	0.63%	-0.20%	-0.24%	-0.26%	-0.18%	-0.30%	-0.03%	-0.50%	-0.46%	-0.29%
E	0.40%	0.26%	0.29%	-0.06%	0.16%	0.09%	-0.13%	-0.10%	0.07%	-0.52%	-0.46%	-0.24%
Avg.	**0.67%**	**0.38%**	**0.51%**	**-0.09%**	**-0.05%**	**-0.11%**	**-0.11%**	**-0.09%**	**0.00%**	**-0.37%**	**-0.14%**	**-0.23%**
D	0.70%	0.67%	0.40%	-0.12%	0.52%	0.12%	-0.09%	0.27%	-0.06%	-0.45%	0.02%	-0.29%

First, we validate the effectiveness of SA-MC and study the influence of different components of SA-MC, including the two-step prediction fusion (TWO-STEP) and overlapped region-based weighted strategy (ORMC). As shown in Table VIII, we introduce two variants for comparison with SA-MC. ONE-STEP indicates that the final prediction is directly derived according to the segmentation mask from the block-level primary and secondary predictions at the end of fractional-part MC. TWO-STEP indicates that prediction fusion is performed on the integer and fractional parts of MC. The integer-part fusion mainly aims to generate the accurate integer pixels for the subsequent fractional MC process, and provides the precise adjacent reference pixels for the interpolation of edge-adjacent fractional pixels. SAIP indicates that the TWO-STEP and ORMC are all used in SA-MC. Through the experimental results, we draw two conclusions: (1) For the comparison of ONE-STEP and SAIP, the experimental results show that SA-MC is superior to the direct multi-region MC, which demonstrates the assistance of a segmentation mask is beneficial to generating accurate prediction. (2) For the comparison of two variants and SAIP, the experimental results show that TWO-STEP and ORMC have a significant effect on SA-MC, and further demonstrate that TWO-STEP can better assist DCTIF in generating accurate fractional-pixel prediction and ORMC is beneficial to reducing the boundary artifacts.

Second, we validate the effectiveness of SA-MVC. In Table VIII, we introduce a variant for comparison with SA-MVC. Instead of using SA-MVC, the secondary candidate list is constructed according to the regular merge candidate order.

For the comparison of variant and SAIP, the experimental results show that our proposed MVC method helps achieve more bits saving than the regular method, which demonstrates the assistance of a segmentation mask is beneficial to coding the MV efficiently.

2) Comparison of Mode Selection: Figure 9 shows the mode selected results of the GEO mode and our proposed SAIP mode, more results are shown in Section III of supplementary material. Compared with GEO, SAIP possesses more flexible motion field description capability, which focuses on more motion-inconsistent regions at the edges of moving objects and handles more difficult cases such as the outer outline of the people in *BQMall*. Meanwhile, the SAIP not only tackles the efficient representation of complex motion fields, but also tends to hit the larger block to save the bit consumption of the refined partition.

D. Limitation and Potential

In this subsection, we discuss the limitations and potential of the proposed SAIP through its effectiveness on different scenes and configurations.

1) Limitation: Based on the above experiments, we find that the performance of SAIP is limited by segmentation accuracy. Here we analyze it in two aspects: (1) *Segmentation Technology*: From the CTC results of SAIP (Table II), we find that our method performs poorly on some sequences with special cases (extreme motion, serious occlusion, large resolution), which is mainly limited by the poor segmentation results, such as *BQTerrence* (camera motion), *Tango2* (indistinguishable foreground and background), *Campfire* (dense moving

TABLE IX
BD-RATE RESULTS OF OUR PROPOSED SAIP METHOD COMPARED TO VTM-12.0 ON THE TEST SEQUENCES OF VOS'S TRAIN AND VALIDATION DATASET [52]–[54], THE DETAILED SETTINGS OF DATASET ARE SHOWN IN SECTION II-B OF SUPPLEMENTARY MATERIAL.

Dataset	Sequence Num	Objects Num[1]	Selected Character	LDB (R^2/L^3) Average[5]	LDB (R^2/L^3) Highest[5]	RA (R/L) Average	RA (R/L) Highest	Accuracy (R-L)[4] $\mathcal{J}$ & $\mathcal{F}$ Mean[4]	LDB Δ[6] Average	LDB Δ[6] Highest	RA Δ[6] Average	RA Δ[6] Highest
DAVIS-17 [52]	30	99	Diverse Scenes	-0.21%/-0.51%	-0.36%/-0.81%	-0.11%/-0.39%	-0.33%/-0.72%	83.2/83.1/81.6/77.7	-0.30%	-0.45%	-0.28%	-0.39%
Youtube-VOS [53]	30	120	Multiple Objects	-0.19%/-0.41%	-0.31%/-0.77%	-0.04%/-0.31%	-0.22%/-0.64%	80.7/79.3/77.3/74.9	-0.22%	-0.46%	-0.27%	-0.42%
MOSE [54]	30	295	Complex Motions	-0.16%/-0.43%	-0.41%/-0.73%	0.09%/-0.27%	-0.25%/-0.53%	72.3/68.5/62.7/57.2	-0.27%	-0.32%	-0.36%	-0.28%
Overall	**90**	**514**	**High Quality**	**-0.19%/-0.45%**	**-0.36%/-0.77%**	**-0.02%/-0.32%**	**-0.27%/-0.63%**	**78.7/77.0/73.9/69.9**	**-0.26%**	**-0.41%**	**-0.30%**	**-0.36%**

[1] Objects Num denotes the total object number of the test sequences. The object number of each test sequence is shown in Section II-B of Supplementary Material.
[2] R indicates the test results are performed by the proposed reference frame-based segmentation method (Section IV).
[3] L indicates the test results are performed by the reference frame's hand-annotated label segmentation mask (open-source annotation of train and validation dataset [52]–[54]).
[4] Accuracy (R-L) indicates the similarity between the proposed segmentation method's result and hand-annotated label mask, which is evaluated by the mean of region similarity $\mathcal{J}$ and contour accuracy $\mathcal{F}$ [38] ($\mathcal{J}_s$ / $\mathcal{F}_s$ in [53]). The four results correspond to the segmentation similarity under different quantization parameters (22, 27, 32, 37).
[5] Average indicates the average BD-rate reduction on the Y Component of proposed method on the test sequences. Highest indicates the highest performance among them.
[6] Δ indicates the SAIP's performance difference between R- and L-based segmentation generation method.
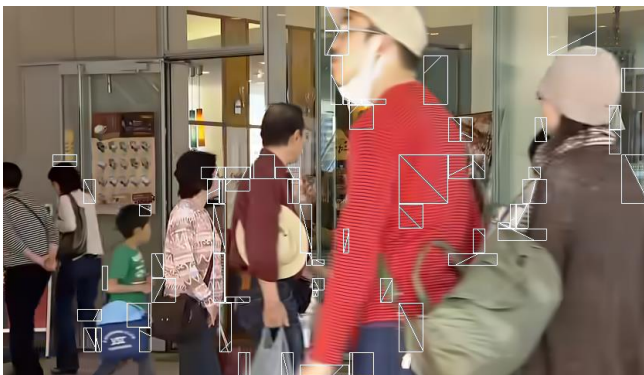
(a) GEO (BQMall, QP: 37, POC : 68)
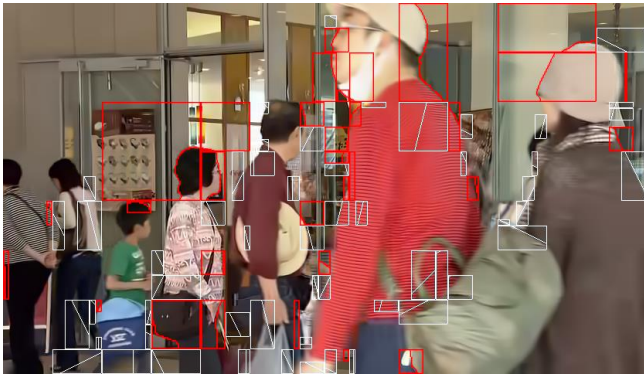
(b) GEO + SAIP (BQMall, QP : 37, POC: 68)

Fig. 9. Mode Selection Results on VTM-12.0 (Configuration: LDB), where (a) uses the GEO mode (anchor: VTM-12.0), (b) uses the GEO mode and proposed SAIP mode. The white/red blocks indicate the block with GEO/SAIP selected. (More results are shown in the supplementary material.)

objects). (These failure segmentation cases are shown in Section II-A of supplementary material.) Meanwhile, learned segmentation schemes have not yet been widely generalized to large-resolution video, such as 1080P, 4k (resolution of Class A, B). The datasets and benchmarks [52], [53] of these segmentation schemes are mainly evaluated on medium resolution (480P-720P). Due to the limited capability and generalization of the learned segmentation method [31], [37], it's difficult to generate accurate segmentation to assist the motion prediction of SAIP in some challenging scenes. (2)

Dependence of Temporal-domain Reference Structure: From the comparison of SAIP's CTC results under different configurations in Table II , we find that the reference structure also affects its performance. Due to the motion area partition of each block being derived from the reference block, the motion area partition of the long-distance reference block is difficult to adapt to the motion situation of the current coding block and further influences prediction accuracy, such as the RA reference structure case.

2) Potential: To explore the potential improvement brought by segmentation accuracy, we test the SAIP on the datasets (train and validation datasets of VOS Benchmark [52]–[54]) with hand-annotated label segmentation mask to avoid the above limitations. For each dataset, we select 30 videos with multiple moving objects, and test the SAIP under different segmentation settings, including the proposed segmentation method (Section IV) and hand-annotated label segmentation mask. The detailed settings and experimental results are shown in Table IX . Table IX shows the difference of SAIP's performance under different segmentation settings (LDB/RA Δ) can achieve up to -0.41%, -0.36%, and on average -0.26%, -0.30% BD-rate reduction (Y component) on VTM-12.0. (The visualization of the different settings' segmentation results is shown in Section II-B of supplementary material.) It demonstrates that the SAIP can achieve higher performance with the assistance of the better segmentation mask, and exhibits positive performance potential with continuous enhancement of advanced segmentation technology [16]–[18] in the future.

X. CONCLUSION

In this paper, object segmentation-assisted inter prediction is proposed to improve the representation of complex motion fields in inter-frame coding. We introduce segmentation information to help the codec achieve the flexible partition of motion area, and further utilize it to assist the entire inter prediction process. We exploit the potential semantic information to improve the coding performance under the advanced coding standard. We have conducted extensive experiments to verify the effectiveness of our method. Experimental results show that the proposed method achieves on average 0.82%, 0.49% and 0.37% BD-rate reduction compared to VTM anchor, under

the LDP, LDB, RA profiles, respectively, and also performs better than the other state-of-the-art methods.

In future work, we will further extend the SAIP with flexible motion modeling in VVC. The design of the current SAIP mainly addresses translation motion in the coding scene. We may further consider the additional prediction process to solve the more complex situation like rotation and zooming. We will further explore the combination of SAIP and the motion model [55]–[59]. In addition, the potential semantic information can also be used to improve the performance of the other modules in the codec, such as quantization [60], [61], transform [62], [63], and post-processing [64], [65]. We will further extend the other modules with the assistance of segmentation information to achieve good performance in future video coding.

REFERENCES

[1] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on circuits and systems for video technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[2] B. Bross, Y.-K. Wang, Y. Ye, S. Liu, J. Chen, G. J. Sullivan, and J.-R. Ohm, "Overview of the versatile video coding (VVC) standard and its applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3736–3764, 2021.

[3] Y.-Q. Zhang and S. Zafar, "Predictive block-matching motion estimation for TV coding. II. Inter-frame prediction," *IEEE transactions on broadcasting*, vol. 37, no. 3, pp. 102–105, 1991.

[4] J.-Y. Lee, J.-K. Han, J.-G. Kim, and T. Q. Nguyen, "Efficient inter-view motion vector prediction in multi-view HEVC," *IEEE Transactions on Broadcasting*, vol. 64, no. 3, pp. 666–680, 2017.

[5] W.-J. Chien, L. Zhang, M. Winken, X. Li, R.-L. Liao, H. Gao, C.-W. Hsu, H. Liu, and C.-C. Chen, "Motion vector coding and block merging in the versatile video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3848–3861, 2021.

[6] H. Yang, H. Chen, J. Chen, S. Esenlik, S. Sethuraman, X. Xiu, E. Alshina, and J. Luo, "Subblock-based motion derivation and inter prediction refinement in the versatile video coding standard," *IEEE transactions on circuits and systems for video technology*, vol. 31, no. 10, pp. 3862–3877, 2021.

[7] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H. 264/AVC video coding standard," *IEEE Transactions on circuits and systems for video technology*, vol. 13, no. 7, pp. 560–576, 2003.

[8] I.-K. Kim, J. Min, T. Lee, W.-J. Han, and J. Park, "Block partitioning structure in the HEVC standard," *IEEE transactions on circuits and systems for video technology*, vol. 22, no. 12, pp. 1697–1706, 2012.

[9] Y.-W. Huang, J. An, H. Huang, X. Li, S.-T. Hsiang, K. Zhang, H. Gao, J. Ma, and O. Chubach, "Block partitioning structure in the VVC standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3818–3833, 2021.

[10] M. Bläser, C. Heithausen, and M. Wien, "Geometry-adaptive motion partitioning using improved temporal prediction," in *2017 IEEE Visual Communications and Image Processing (VCIP)*. IEEE, 2017, pp. 1–4.

[11] H. Gao, S. Esenlik, E. Alshina, and E. Steinbach, "Geometric partitioning mode in versatile video coding: Algorithm review and analysis," *IEEE transactions on circuits and systems for video technology*, vol. 31, no. 9, pp. 3603–3617, 2020.

[12] J. Chen, S. Lee, K.-H. Lee, and W.-J. Han, "Object boundary based motion partition for video coding," in *Picture Coding Symposium*, 2007.

[13] M. Bläser, C. Heithausen, and M. Wien, "Segmentation-based partitioning for motion compensated prediction in video coding," in *2016 Picture Coding Symposium (PCS)*. IEEE, 2016, pp. 1–5.

[14] Z. Wang, S. Wang, X. Zhang, S. Wang, and S. Ma, "Three-zone segmentation-based motion compensation for video compression," *IEEE Transactions on Image Processing*, vol. 28, no. 10, pp. 5091–5104, 2019.

[15] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 7, pp. 3523–3542, 2021.

[16] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4015–4026.

[17] T. Zhou, F. Porikli, D. J. Crandall, L. Van Gool, and W. Wang, "A survey on deep learning technique for video segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 6, pp. 7099–7122, 2022.

[18] Y. Cheng, L. Li, Y. Xu, X. Li, Z. Yang, W. Wang, and Y. Yang, "Segment and track anything," *arXiv preprint arXiv:2305.06558*, 2023.

[19] H. Gao *et al.*, "Integrated text for GEO," JVET, Tech. Rep. JVET-Q0806, 2016.

[20] X. Meng, C. Jia, X. Zhang, S. Wang, and S. Ma, "Spatio-temporal correlation guided geometric partitioning for versatile video coding," *IEEE Transactions on Image Processing*, vol. 31, pp. 30–42, 2021.

[21] P. Salembier, F. Marqués, M. Pardas, J. R. Morros, I. Corset, S. Jeannin, L. Bouchard, F. Meyer, and B. Marcotegui, "Segmentation-based video coding system allowing the manipulation of objects," *IEEE transactions on circuits and systems for video technology*, vol. 7, no. 1, pp. 60–74, 1997.

[22] T. Meier and K. N. Ngan, "Video segmentation for content-based coding," *IEEE transactions on circuits and systems for video technology*, vol. 9, no. 8, pp. 1190–1203, 1999.

[23] X. Zheng, H. Yu, S. Li, Y. He, and P. Bordes, "CE2: non-rectangular motion partitioning," in *Joint Collaborative Team on Video Coding (JCTVC) of ITU-T VCEG and ISO/IEC MPEG, Torino, Itlay, 6th meeting, Tech. Rep. JCTVC-F415*, 2011.

[24] M. Blaser, J. Sauer, and M. Wien, "CE10: Results on Geometric Block Partitioning (Test 3.3), JVET, Document JVET-K0146," in *11th Meeting, Ljubljana, SI*, vol. 2, 2018.

[25] M. Pardas, P. Salembier, F. Marqués, and R. Morros, "Partition tree for a segmentation-based video coding system," in *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, vol. 4. IEEE, 1996, pp. 1982–1985.

[26] M. T. Orchard, "Predictive motion-field segmentation for image sequence coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 3, no. 1, pp. 54–70, 1993.

[27] J. H. Kim, A. Ortega, P. Yin, P. Pandit, and C. Gomila, "Motion compensation based on implicit block segmentation," in *2008 15th IEEE International Conference on Image Processing*. IEEE, 2008, pp. 2452–2455.

[28] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[29] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[30] Z. Cai and N. Vasconcelos, "Cascade r-cnn: High quality object detection and instance segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 5, pp. 1483–1498, 2019.

[31] T. Vu, H. Kang, and C. D. Yoo, "Scnet: Training inference sample consistency for instance segmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 3, 2021, pp. 2701–2709.

[32] 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 transactions on pattern analysis and machine intelligence*, vol. 41, no. 6, pp. 1515–1530, 2018.

[33] Z. Yang and Y. Yang, "Decoupling features in hierarchical propagation for video object segmentation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 36 324–36 336, 2022.

[34] Y. Xu, Z. Yang, and Y. Yang, "Integrating boxes and masks: A multi-object framework for unified visual tracking and segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 9738–9751.

[35] Y.-T. Hu, J.-B. Huang, and A. G. Schwing, "Videomatch: Matching based video object segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 54–70.

[36] S. W. Oh, J.-Y. Lee, N. Xu, and S. J. Kim, "Video object segmentation using space-time memory networks," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9226–9235.

[37] H. K. Cheng, Y.-W. Tai, and C.-K. Tang, "Rethinking space-time networks with improved memory coverage for efficient video object segmentation," *Advances in Neural Information Processing Systems*, vol. 34, pp. 11 781–11 794, 2021.

[38] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 724–732.

[39] C. Auyeung, J. J. Kosmach, M. T. Orchard, and T. Kalafatis, "Over-lapped block motion compensation," in *Visual Communications and Image Processing'92*, vol. 1818. SPIE, 1992, pp. 561–572.

[40] M. T. Orchard and G. J. Sullivan, "Overlapped block motion compensation: An estimation-theoretic approach," *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 693–699, 1994.

[41] J. M. Prewitt *et al.*, "Object enhancement and extraction," *Picture processing and Psychopictorics*, vol. 10, no. 1, pp. 15–19, 1970.

[42] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[43] H. Lv, R. Wang, X. Xie, H. Jia, and W. Gao, "A comparison of fractional-pel interpolation filters in HEVC and H. 264/AVC," in *2012 Visual Communications and Image Processing*. IEEE, 2012, pp. 1–6.

[44] K. Ugur, A. Alshin, E. Alshina, F. Bossen, W.-J. Han, J.-H. Park, and J. Lainema, "Motion compensated prediction and interpolation filter design in H. 265/HEVC," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 6, pp. 946–956, 2013.

[45] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu *et al.*, "MMDetection: Open mmlab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019.

[46] K. Suehring and X. Li, "JVET common test conditions and software reference configurations," JVET, Tech. Rep. JVET-G1010, 2017.

[47] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," ITU-T VCEG, Tech. Rep. VCEG-M33, 2001.

[48] K. McCann *et al.*, "High efficiency video coding (HEVC) test model 14 (HM 14) encoder description," JCTVC, Tech. Rep. JCTVC-P1002, 2014.

[49] B. Bross, "Versatile video coding (draft 1)," JVET, Tech. Rep. JVET-J1001, 2018.

[50] F. Bossen *et al.*, "Common test conditions and software reference configurations," *JCTVC-L1100*, vol. 12, no. 7, p. 1, 2013.

[51] Xiph Video Test Media, "https://media.xiph.org/video/derf/."

[52] 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," *arXiv preprint arXiv:1803.00557*, 2018.

[53] N. Xu, L. Yang, Y. Fan, J. Yang, D. Yue, Y. Liang, B. Price, S. Cohen, and T. Huang, "Youtube-vos: Sequence-to-sequence video object segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 585–601.

[54] H. Ding, C. Liu, S. He, X. Jiang, P. H. Torr, and S. Bai, "MOSE: A new dataset for video object segmentation in complex scenes," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 20 224–20 234.

[55] H. Watanabe and S. Singhal, "Windowed motion compensation," in *Visual Communications and Image Processing'91: Visual Communication*, vol. 1605. SPIE, 1991, pp. 582–589.

[56] L. Li, H. Li, D. Liu, Z. Li, H. Yang, S. Lin, H. Chen, and F. Wu, "An efficient four-parameter affine motion model for video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 8, pp. 1934–1948, 2017.

[57] J. Wang, D. Wang, and W. Zhang, "Temporal compensated motion estimation with simple block-based prediction," *IEEE Transactions on Broadcasting*, vol. 49, no. 3, pp. 241–248, 2003.

[58] H. Gao, X. Chen, S. Esenlik, J. Chen, and E. Steinbach, "Decoder-side motion vector refinement in VVC: Algorithm and hardware implementation considerations," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 8, pp. 3197–3211, 2020.

[59] Y. Li, Z. Li, L. Li, D. Liu, and H. Li, "Global homography motion compensation for versatile video coding," in *2022 IEEE International Conference on Visual Communications and Image Processing (VCIP)*. IEEE, 2022, pp. 1–5.

[60] H. Schwarz, M. Coban, M. Karczewicz, T.-D. Chuang, F. Bossen, A. Alshin, J. Lainema, C. R. Helmrich, and T. Wiegand, "Quantization and entropy coding in the versatile video coding (VVC) standard," *IEEE Transactions on circuits and systems for video technology*, vol. 31, no. 10, pp. 3891–3906, 2021.

[61] M. Xu and B. Jeon, "Complexity-efficient dependent quantization for versatile video coding," *IEEE transactions on broadcasting*, vol. 69, no. 3, pp. 832–839, 2023.

[62] X. Zhao, S.-H. Kim, Y. Zhao, H. E. Egilmez, M. Koo, S. Liu, J. Lainema, and M. Karczewicz, "Transform coding in the VVC standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3878–3890, 2021.

[63] T. L. da Silveira, D. R. Canterle, D. F. Coelho, V. A. Coutinho, F. M. Bayer, and R. J. Cintra, "A class of low-complexity dct-like transforms for image and video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 7, pp. 4364–4375, 2021.

[64] M. Karczewicz, N. Hu, J. Taquet, C.-Y. Chen, K. Misra, K. Andersson, P. Yin, T. Lu, E. François, and J. Chen, "VVC in-loop filters," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3907–3925, 2021.

[65] C. Chen, Z. Miao, and B. Zeng, "Adaptive guided image filter for improved in-loop filtering in video coding," in *2015 IEEE 17th International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 2015, pp. 1–6.

Supplementary Material for
Object Segmentation-Assisted Inter Prediction for Versatile Video Coding

Zhuoyuan Li, Zikun Yuan, Li Li, Dong Liu, Xiaohu Tang, and Feng Wu

I. Learned Segmentation for Reference Pictures

Here we show some segmentation masks of CTC test sequences (Fig. 1) segmented by our proposed segmentation strategy in the actual coding process.

(A) BQMall, Poc:76 — Proposed Method, Reconstructed Frame QP: 32 — (B) BaketballDrill, Poc:5 — Proposed Method, Reconstructed Frame QP: 27

(C) MarketPlace, Poc:9 — Proposed Method, Reconstructed Frame QP: 32 — (D) BasketballDrive, Poc:58 — Proposed Method, Reconstructed Frame QP: 27
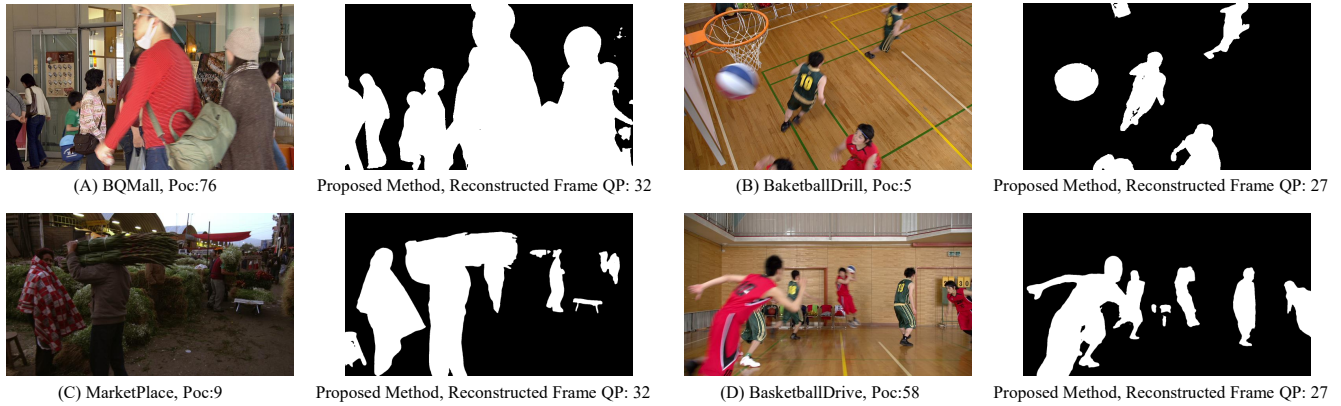
Fig. 1. Visualization of segmentation masks of CTC test sequences segmented by our proposed segmentation strategy in the actual coding process.

II. Limitation and Potential

A. Visualization of Failure Segmentation Cases

Supplement Section IX.D 1), here we show some failure segmentation cases of CTC test sequences in Fig. 2.

(A) BQTerrace, Poc:62 — Failure Segmentation Case: Camera Motion with Tiny Objects — (B) Tango2, Poc:47 — Failure Segmentation Case: Indistinguishability of Foreground and Background

(C) Campfire, Poc:34 — Failure Segmentation Case: Dense Moving Objects — (D) RitualDance, Poc:83 — Failure Segmentation Case: Camera Motion with Dense Objects

Fig. 2. Failure segmentation cases on some sequences with special cases (extreme motion, serious occlusion, large resolution).

B. Setting of Potential Exploration Experiments

Supplement Table IX (BD-rate results of our proposed SAIP method compared to VTM-12.0 on the test sequences of VOS's train and validation dataset), here we detail the setting of the selected test sequences of VOS's train and validation dataset. For

(A) *Davis-17*, gold-fish, Poc:7 — Hand-annotated Label Mask — Proposed Segmentation Method, QP:37

(B) *Youtube-VOS*, Poc:10 — Hand-annotated Label Mask — Proposed Segmentation Method, QP:32

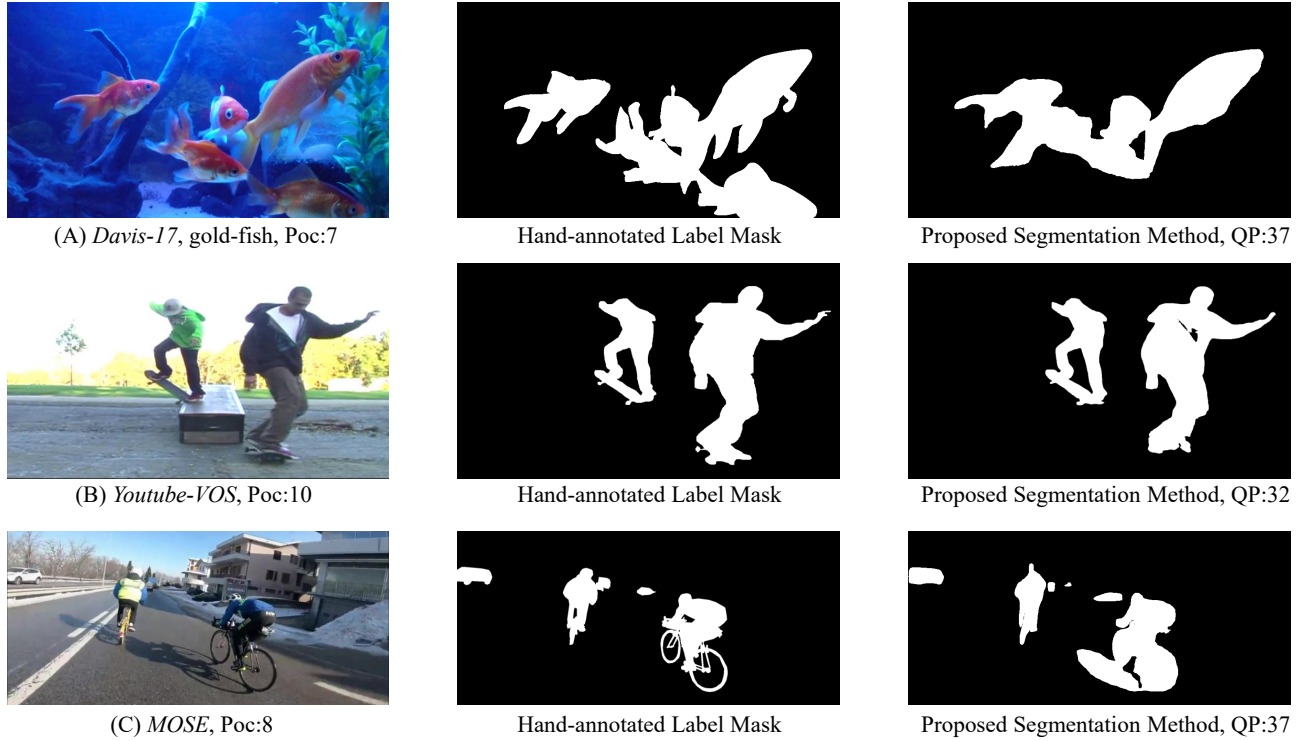(C) *MOSE*, Poc:8 — Hand-annotated Label Mask — Proposed Segmentation Method, QP:37

Fig. 3. The difference of segmentation results of some selected test sequences of *Davis-17* [1], *Youtube-VOS* [2], *MOSE* [3].

each benchmark (*Davis-17* [1], *Youtube-VOS* [2], *MOSE* [3]), we select 30 videos with multiple moving objects. The selected sequences and the number of objects of each sequence (in parentheses) are shown as follows.

Davis-17 [1] (99 objects in total): bike-packing (2), boxing-fisheye (3), cat-girl (2), classic-car (3), color-run (3), crossing (3), dancing (3), disc-jockey (3), dogs-jump (3), dogs-scale (4), drone (4), gold-fish (5), horsejump-high (2), horsejump-low (2), india (3), judo (2), kid-football (2), lab-coat (3), lindy-hop (8), loading (3), longboard (5), motocross-bumps (2), motocross-jump (2), night-race (2), pigs (3), schoolgirls (7), sheep (5), shooting (3), tuk-tuk (3), train (4).

Youtube-VOS [2] (120 objects in total): 0a7b27fde9 (2), 0a8c467cc3 (3), 0a23765d15 (4), 0ce06e0121 (2), 0043f083b5 (3), 00917dcfc4 (3), 011ac0a06f (5), 01c4cb5ffe (3), 01c76f0a82 (4), 0348a45bca (5), 0358b938c1 (4), 03c95b4dae (3), 07353b2a89 (4), 077d32af19 (4), 07c62c3d11 (3), 1122c1d16a (6), 1250423f7c (4), 165c42b41b (4), 19e061eb88 (4), 1aa3da3ee3 (4), 1af8d2395d (4), 1d3087d5e5 (5), 1ef8e17def (3), 1f7d31b5b2 (3), 1f8014b7fd (4), 21f4019116 (5), 351cfd6bc5 (5), 4122aba5f9 (5), c55784c766 (6), c4f5b1ddcc (6).

MOSE [3] (295 objects in total): 78a1f4b1 (5), 55ae6921 (4), 072e7b3f (4), 23b9a3ea (14), a386eaa2 (6), 905d8740 (6), 93e22d48 (10), 25af2eeb (20), 6da17988 (8), fbfb6e30 (12), f6f8b4a6 (9), 662a1846 (9), 37af9ac1 (10), 8d0d641a (9), 831e0009 (8), 8c8b7913 (9), ba538072 (13), fabae187 (10), 82667d52 (10), ba0e4dd4 (8), 7be91184 (11), 14655f54 (10), adf443e1 (11), 2f477d05 (13), 2a2745e5 (11), e638e950 (13), 255f86ef (12), 86a2b59f (20), 23e046fb (10), 06ea172e (12).

Here we show the difference of segmentation results of some selected test sequences of *Davis-17* [1], *Youtube-VOS* [2], *MOSE* [3] in Fig. 3 , including the results segmented by our proposed segmentation method, hand-annotated label segmentation mask provided by the open-source annotation of train and validation dataset.

III. Mode Selection Results

In Fig. 9, we show the selection results of different modes (GEO, SAIP) on the BQMall sequence. Here we show more selection results on other sequences of CTC for the comparison of different modes (GEO, SAIP) in Fig. 4 .

References

[1] 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," *arXiv preprint arXiv:1803.00557*, 2018.

[2] N. Xu, L. Yang, Y. Fan, J. Yang, D. Yue, Y. Liang, B. Price, S. Cohen, and T. Huang, "Youtube-vos: Sequence-to-sequence video object segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 585–601.

[3] H. Ding, C. Liu, S. He, X. Jiang, P. H. Torr, and S. Bai, "MOSE: A new dataset for video object segmentation in complex scenes," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 20 224–20 234.
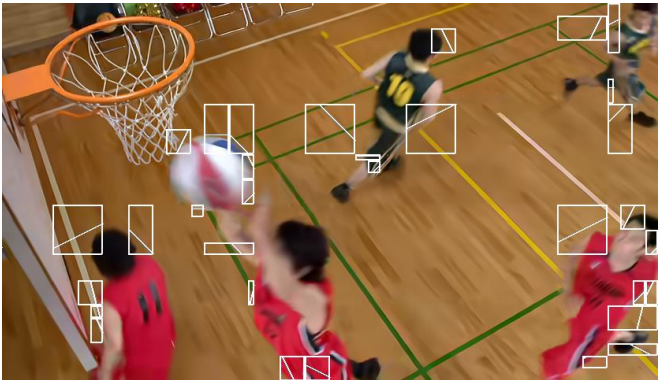
(a) GEO (Johnny, QP=37, POC=83)
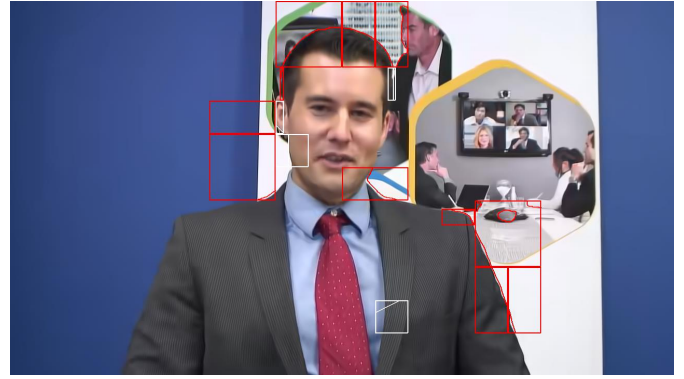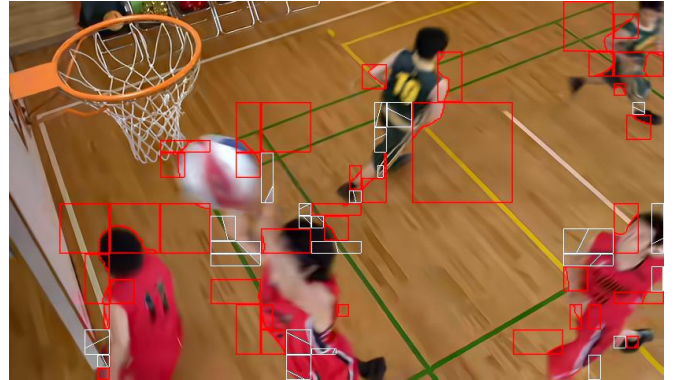
(b) GEO + SAIP (Johnny, QP=37, POC=83)

(a) GEO (MarketPlace, QP=32, POC=10)

(b) GEO + SAIP (MarketPlace, QP=32, POC=10)

(a) GEO (BasketballDrill, QP=37, POC=28)

(b) GEO + SAIP (BasketballDrill, QP=37, POC= 28)

(a) GEO (BlowingBubbles, QP=37, POC=56)

(b) GEO + SAIP (BlowingBubbles, QP=37, POC=56)

Fig. 4. The selection results of different modes of CTC test sequences on VTM-12.0 (Configuration: LDB), where (a) uses the GEO mode (anchor: VTM-12.0), (b) uses the GEO mode and proposed SAIP mode. The white/red blocks indicate the block with GEO/SAIP selected.