IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 7 July 2022, accepted 23 July 2022, date of publication 27 July 2022, date of current version 4 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3194314

RESEARCH ARTICLE

Enhanced Intra String Copy for Screen Content Coding in AVS3

LIPING ZHAO[1,2], (Member, IEEE), HUIHUI WANG[3], QINGYANG ZHOU[3], WENJUAN ZHANG[3], KAILUN ZHOU[3], KELI HU[1,2], (Member, IEEE), AND TAO LIN[3]

[1]Department of Computer Science and Engineering, Shaoxing University, Shaoxing 312000, China
[2]Information Technology Research and Development Innovation Center, Peking University, Shaoxing 312000, China
[3]VLSI Laboratory, College of Electronics and Information Engineering, Tongji University, Shanghai 200092, China

Corresponding author: Tao Lin (lintao@tongji.edu.cn)

This work was supported in part by the Natural Science Foundation of Zhejiang Province under Grant LTY22F020003, Grant LY20F020011, and Grant TY22F025548; in part by the National Science Foundation of China under Grant 61871289; in part by the Natural Science Foundation of Shanghai under Grant 19ZR1461100; and in part by the Social Sciences and Humanities Youth Foundation of Ministry of Education of China under Grant 21YJCZH039.

ABSTRACT Driven by growing applications that use computer screens as interfaces for daily remote interactions, almost all current video coding standards have included screen content coding (SCC) tools. Recently, an efficient SCC tool called intra string copy (ISC) has been adopted in the third-generation of audio video coding standard in China (AVS3). ISC has two coding unit (CU) level sub-modes: fully-matching-string and partially-matching-string based string prediction (FPSP) sub-mode and equal-value-string, unit-basis-vector-string, and unmatched-pixel-string based string prediction (EUSP) sub-mode. To further improve the coding efficiency of SCC, this paper proposes four enhancement techniques of ISC (EISC), including CU partition improvements, point vector (PV) relocation and reactivation, line-based overlapping string prediction, and an optimized coding method for string length in the EUSP sub-mode. Compared with the latest AVS3 reference software HPM with EISC disabled, using AVS3 SCC common test condition and YUV test sequences in text and graphics with motion and mixed content categories, the proposed technique achieves an average Y BD-rate reduction of 2.39% and 1.49% for all intra (AI) and low-delay B (LDB) configurations, respectively, with low additional encoding complexity and almost no additional decoding complexity. All proposed ISC enhancement techniques have been adopted in AVS3.

INDEX TERMS Video coding standard, screen content coding, intra string copy, string split, point vector.

I. INTRODUCTION

With more and more popular applications using computer screen as interfaces for daily remote human-machine and human-human interactions, such as online education, teleconferencing, telecommuting, and remote entertainment, almost all recent video coding standards have included screen content coding (SCC) tools [1]. The video coding standards with SCC tools include high efficiency video coding (HEVC), versatile video coding (VVC) [2], the second-generation audio video standard (AVS2), the third-generation AVS (AVS3) [3], [4], AV1, and essential video coding (EVC).

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

In various standards, the adopted SCC tools are significantly different [1]. Five main SCC tools have been adopted in VVC [2]: transform skip residual coding, block-based differential pulse-code modulation (BDPCM), intra-block copy (IBC), adaptive color transform (ACT), and palette. In AVS3, five main SCC tools have been adopted in the draft [5]: IBC [6], frequency-based intra-mode coding (FIMC) [7], implicit selection of transform skip (ISTS) [8], adaptive control of deblocking type (ACDT) [9], and intra string copy (ISC) [10]–[12].

Among these SCC tools, ISC has been included in AVS2 [13] and AVS3 [10]–[12]. ISC, also known as string prediction (SP) or string matching (SM), has several early versions, including macro-block or coding tree unit (CTU)-based 1D SM [14]–[17], pseudo 2D SM [18]–[19], index-based

SM [20], universal string matching (USM) implemented in HEVC [21]–[22], and USP in AVS2 [12], [23]–[24]. The basic idea of ISC is to divide a coding unit (CU) into multiple strings to take full advantage of matching patterns with a variety of sizes, shapes, and positions in screen content.

In AVS3, a basic version of the ISC was adopted in the early stages of AVS3 development [10]–[12]. The ISC has two CU level sub-modes: fully-matching-string (Fms) and partially-matching-string (Pms) based string prediction (FPSP) sub-mode and equal-value-string (Evs), unit-basis-vector-string (Ubvs), and unmatched-pixel-string (Ups) based string prediction (EUSP) sub-mode. The major features of the basic ISC version include 4:2:0 chroma format supported ISC with various CU partitions, optimal string searching strategies to search for matching strings in the reference range, efficient coding methods for string parameters, and a variety of string constraints to reduce hardware implementation complexity.

SCC is increasingly used in mobile devices, and low hardware implementation complexity becomes as important as high coding efficiency when developing video coding standards. Improving the coding efficiency with low hardware implementation complexity is a new challenge. The basic version of the ISC adopted by AVS3 has a relatively low complexity without exploring the full coding efficiency potential of the ISC. To fill this gap, we propose four enhancement techniques in three major string coding functional modules of ISC (EISC), as follows:

1) **One improvement technique in the CU partitions coding module**: The CU partition of the EUSP sub-mode extends to Y-only (Tree L) CUs and small CUs with a width and/or height equal to four pixels. A well-selected total number of PVs used in small CUs is proposed to maintain a balance between coding efficiency and low complexity. Novel strategies are implemented to handle the differences between the Y CUs and YUV CUs of the EUSP sub-mode.
2) **Two optimization techniques in the string prediction module**: a) Novel point vector (PV) relocation and PV reactivation strategies that move or resume PVs to or in new positions (PVs) with the same frequently occurring position (FOP) pixel values are proposed. The new PVs after PV relocation or reactivation stay much longer in the valid reference range; thus, a higher coding efficiency is obtained. b) Based on the particularity of a special type of overlapping string, a line-based overlapping string prediction method is proposed to achieve high coding efficiency by splitting a single overlapping string into multiple nonoverlapping lines. To maintain low complexity, the reference area of the PV or string vector (SV) is exactly the same as that of the basic version of the ISC.
3) **One optimization technique in the string parameter coding module**: It is observed that the statistical characteristics of the string length for Ubvs are quite different from those of Evs and Ups; thus, an optimized string length parameter coding method is presented to further improve the coding efficiency.

TABLE 1. Notations used in this paper.

Notation	Abbrevia-tion	Description
FPSP	FPSP	A fully-matching-string and partially-matching-string based string prediction sub-mode.
EUSP	EUSP	An equal-value-string, unit-basis-vector-string, and unmatched-pixel-string based string prediction sub-mode.
fully-matching-string	Fms	An Fms is a string without any unmatched pixel.
partially-matching-string	Pms	A Pms is a string with at least one unmatched pixel.
equal-value-string	Evs	A string with all of its reconstructed pixel values equal to each other.
unit-basis-vector-string	Ubvs	A string with string vector equal to (0, -1).
unpredictable-pixel string	Ups	A string with all of its pixels are unpredictable pixels.
string vector	SV	An SV is used to represent the coordinate offset between the current string and the reference string.
string length	Length	A Length is used to represent the length of a string (1~1024).
frequently occurring position	FOP	A group of frequently occurring pixel positions with the same reconstructed pixel value is denoted as the FOP pixel value.
point vector	PV or Pv	A two-dimensional vector specifies the position coordinates of an FOP pixel value used as the reference pixel for all pixels in an Evs.
point vector information list	PvInfoList	A PV information list for all PVs used in a CU. Each entry consists of coordinates and the status of a PV.
historical PV information list	HistPvInfo List	A table of historical and consecutive PvInfoLists of previous EUSP Cus.

All proposed enhancement techniques were adopted in AVS3. This paper was partially based on AVS documents [25]–[29].

The remainder of this paper is organized as follows. Section II describes the related work and background. Section III presents the architecture of the enhanced intra string copy coding system architecture in AVS3. Section IV elaborates the technical details of the proposed EISC. Experimental results are provided and discussed in Section V. Compared with the AVS3 reference software HPM [30] with EISC disabled, for the sequences from the AVS3 SCC Common Test Condition (CTC) [31], the proposed EISC can achieve a significant Bjøntegaard delta rate (BD-rate) [32] reduction with low additional encoding complexity and almost no additional decoding complexity. Conclusions and future work are presented in Section VI.

II. RELATED WORK AND BACKGROUND

Table 1 lists and describes the main notations and abbreviations used in this paper.

As shown in Figure 1, the ISC has two CU level sub-modes: the FPSP sub-mode and the EUSP sub-mode. In both

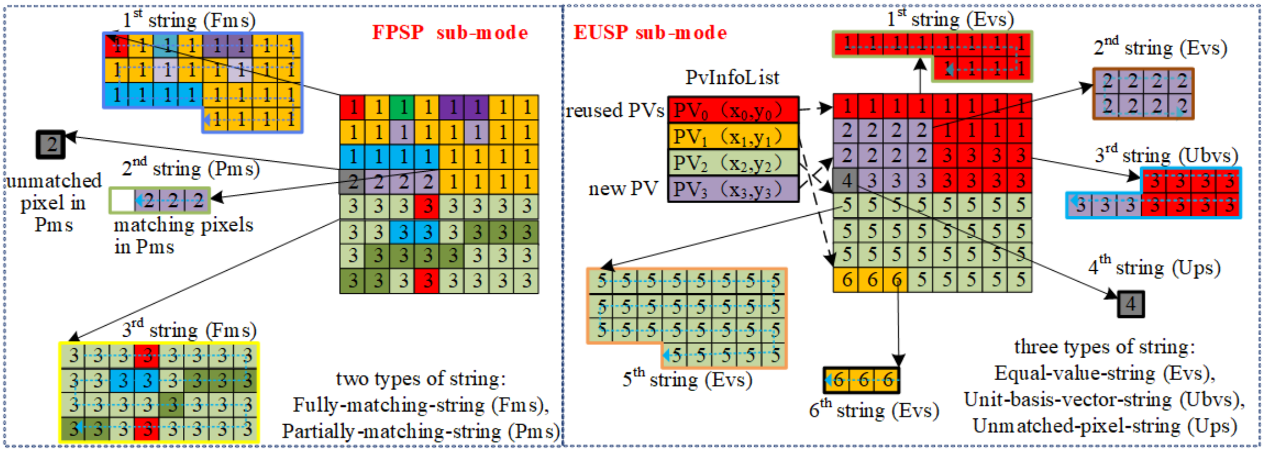


FIGURE 1. Examples of FPSP sub-mode and EUSP sub-mode of ISC.

sub-modes, a CU is divided into multiple strings connected one by one in the order of a horizontal traverse-scan.

In the FPSP sub-mode, there are two types of strings: fully matching string (Fms) and partially matching string (Pms). The left part of Figure 1 illustrates an 8 × 8 CU divided into two Fmss (the 1st and the 3rd strings) and one Pms (the 2nd string). A Pms is a string with at least one unmatched pixel, such as the second string with one unmatched pixel (with a black bounding box marked with black '2') and three matching pixels (with green bounding box and marked with black '2'). An Fms is a string without any unmatched pixels. For each Fms or Pms with at least one matching pixel, a string vector (SV) is used to represent the coordinate offset between the current and reference strings. The length of any Pms is always four pixels, whereas the length of any Fms is always a multiple of four pixels in AVS3. For an unmatched pixel, its pixel value is coded directly into the bitstream.

In the EUSP sub-mode, there are three types of strings: equal-value-string (Evs), unit-basis-vector-string (Ubvs), and unmatched-pixel string (Ups). An example of an 8 × 8 CU divided into four Evses (the 1st, 2nd, 5th, and 6th strings), one Ubvss (the 3rd string), and one Ups (the 4th string) is illustrated in the right part of Figure 1. An Evs is a string with all of its reconstructed pixel values equal to each other, that is, equal to the same reference pixel value called the FOP pixel value. Thus, any Evs can be represented and coded by a PV pointing to an FOP in the reference range. In fact, the value of a PV is the coordinate of an FOP within the reference range. All PVs of the CU are stored in its PvInfoList. PvInfoList generally consists of two parts: reused and new PVs. A reused PV stored in the HistPvInfoList has already been used in previous EUSP CUs and points to a position outside of the current CU, whereas a new PV occurs in the current CU for the first time and points to a position inside the current CU. For an Evs, two syntax elements, pv_address and fop_pixel_val, are coded into the bitstream. pv_address specifies the address of a PV in PvInfoList. fop_pixel_val specifies the FOP pixel value of the new PV. A Ubvs is a matching string with SV equal to $(0, -1)$ and reference pixels inside the current CU. A Ups is a string of unmatched pixels coded directly into a bitstream. The lengths of Evs, Ubvs, and Ups are also coded into the bitstream.

The basic version of ISC in AVS3 has the following major features.

1) 4:2:0 format ISC has been developed. In the FPSP sub-mode, both the YUV CU and the Y CU are supported. In the YUV CU, both luma and chroma share the same string prediction parameters (Length and SV). In the Y CU, the FPSP sub-mode is applied only to the luma component. The FPSP sub-mode is supported by a CU with both width and height smaller than 64 [11]. In the EUSP sub-mode, only the YUV CU with both width and height greater than 4 but smaller than 64 is supported [12].
2) Minimizing ISC hardware implementation complexity. The ISC has the same reference range constraint as the IBC [30], [33]. The ISC uses only the horizontal traverse-scan mode. In the FPSP sub-mode, only nonoverlapping strings (i.e., without overlap between the current string and its reference string) are allowed. The total number of strings in a CU cannot exceed a quarter of the total number of CU pixels [10]. In the EUSP sub-mode, the total number of PVs used in a CU does not exceed 10 for 8 × 8 CU and 15 for CUs of other sizes. The maximum number of PVs stored in the HistPvInfoList (Historical PV Information List) is 28 [12].
3) In the FPSP sub-mode, both SV and Length coding parameters are coded in a highly efficient manner [11]. In the EUSP sub-mode, a PV instance classification-based FOP pixel value coding scheme and an efficient coding scheme for PV addresses were proposed [12].

However, the basic version of the ISC adopted by AVS3 has a relatively low coding complexity without exploring the full coding efficiency potential of the ISC. The basic version of the ISC has the following aspects that can be improved:

1) The EUSP sub-mode does not support the Y CUs or CUs with a width and/or height equal to four pixels.
2) Some constraints on hardware implementation complexity and cost reduction have a significant negative impact on coding efficiency. For example, the reference range of PV is the same as the block vector (BV) of IBC and SV of ISC, and only nonoverlapping strings are allowed in the FPSP sub-mode.
3) The coding efficiency of the ISC coding parameter can be optimized. It is observed that the statistical characteristics of the string length for Ubvs are quite different from those of Evs and Ups, but the same string length coding method is used for all three types of strings without dedicated optimization.

To further improve coding efficiency with low hardware implementation complexity, we propose four enhancement techniques for ISC.

III. ENHANCED ISC CODING SYSTEM ARCHITECTURE

Figure 2 illustrates the architecture of the enhanced ISC coding system. The enhancement techniques of ISC are marked with an orange bounding box in the three main functional modules: CU partition, string prediction, and string parameter coding.

In the CU partition module, the EUSP sub-mode is extended to Y CUs or CUs with width and/or height equal to four pixels. Some unique problems must be solved when the EUSP sub-mode is extended to such CUs. One problem is how to handle the chroma components of the new PVs and the associated FOP pixel values in Y CUs. Another problem is that the total number of PVs used in a smaller CU must be further reduced. The details of ***CU partition improvements*** are provided in ***Section IV.A***.

In the string prediction module, two improvements are added to the EUSP and FPSP sub-modes. In the EUSP sub-mode, a PV relocation and reactivation technique is proposed to allow the PV to stay longer in the reference range for use by more EUSP CUs. PV relocation involves moving a valid reference pixel position pointed to by a PV to a new position with exactly the same reference pixel value by periodically updating the value of the PV. The new position is in the current CU, whereas the previous position is usually outside the reference range. In this way, the PV points to a valid reference pixel position with exactly the same reference pixel value for a longer period. PV reactivation is used when PV relocation fails, and a PV has just exceeded the valid reference range and becomes an invalid PV. Because the FOP pixel value of an invalid PV may be used by a Future EUSP CU, an invalid PV is not deleted from HistPvInfoList as long as HistPvInfoList still has a slot for it. An invalid PV stored in the HistPvInfoList is called a sleeping PV. The associated FOP pixel value of a sleeping PV is temporarily stored in the SlpPVbuf (Sleeping PV buffer). When a sleeping PV is used by a future EUSP CU, it is reactivated and treated as a new PV, but its associated FOP pixel value is obtained from SlpPVbuf instead of from the bitstream. Thus, a sleeping PV can become a valid PV again and can be used by more EUSP CUs. The details of ***PV relocation and reactivation*** are given in ***Section IV.B***.

Another improvement in the string-prediction module is the line-based overlapping string prediction technique. An overlapping string includes an overlap between the current string and its reference string. In fact, an overlapping string is multiple nonoverlapping short strings with the same SV merged together and can be split into multiple nonoverlapping short strings with the same SV. It is obvious that multiple nonoverlapping short strings require more bits to code, resulting in a lower coding efficiency than a single long overlapping string. However, a single string copy of an overlapping string must always be split into multiple string copies of many nonoverlapping short strings (even one pixel). Thus, a general pixel-based overlapping string copy incurs a high hardware implementation cost. However, line-based overlapping string prediction can achieve both high coding efficiency and low hardware implementation cost by splitting a single string copy of an overlapping string into multiple nonoverlapping line copies, which have much lower hardware implementation costs than pixel copies. The details of ***line-based overlapping string prediction*** are provided in ***Section IV.C***.

In the string parameter coding module, to improve the Length coding efficiency, the Length coding method is optimized for all three types of strings by adding an optimized coding and binarization scheme dedicated for Ubvs. The details of ***optimized coding method for Length*** are provided in ***Section IV.D***.

IV. TECHNICAL DETAILS OF EISC

A. CU PARTITION IMPROVEMENTS OF ISC MODE

The improvements of ISC mode CU partition include the following two aspects.

First, a small CU with width and/or height equal to four pixels is supported in the EUSP sub-mode. To minimize the hardware implementation complexity, the total number of PVs used in a CU cannot exceed 2 for 4×4 CUs, 5 for 4×8 and 8×4 CUs, 10 for 8×8, 4×16, 16×4 CUs, and 15 for CUs with $8<\text{width}\leq 32$ and $8<\text{height}\leq 32$.

Second, Y CUs are supported in the EUSP sub-mode. In AVS3, the luma component can split further, whereas the chroma component no longer splits. An example of a Y CU coded by the EUSP sub-mode in I slice is shown in Figure 3. In a 16×16 partition, the luma component is split into four 8×8 CUs, as shown in Figure 3, and coded by the intra-mode, EUSP sub-mode, FPSP sub-mode, and IBC mode. The corresponding chroma component is not split and is coded as a single 8×8 UV CU of the intra mode.

There are three major differences between the EUSP Y CU and YUV CU. The first is that in an Evs of a Y CU, only Y

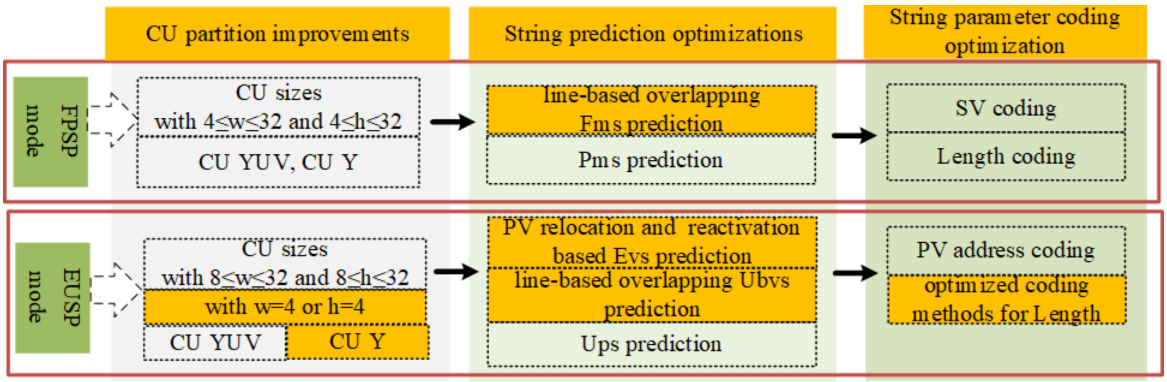


FIGURE 2. Enhanced Intra String Copy coding system architecture.

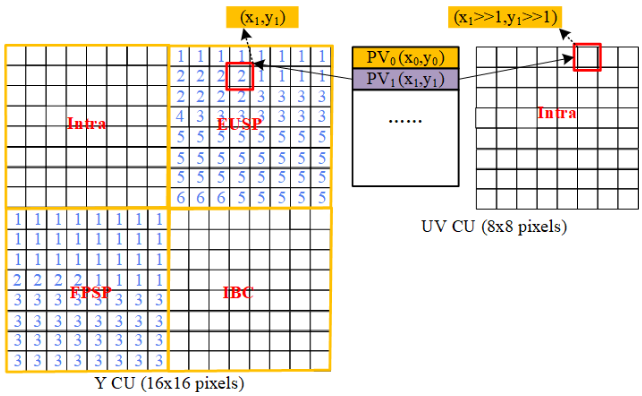


FIGURE 3. An example of EUSP sub-mode coded Y CU in I slice.

samples of all reconstructed pixels are equal to each other. Second, for a new PV in a Y CU, only the luma component of its associated FOP pixel value exists in the bitstream and its corresponding chroma component always comes from the reconstructed UV CU coded by the intra- or inter-mode. As shown in Figure 3, if a new PV denoted by PV_1 is at position (x_1, y_1) of the luma sample array, then the chroma component of its associated FOP pixel value comes from position $(x_1 \gg 1, y_1 \gg 1)$ in the intra-mode reconstructed UV CU. Third, in a Y CU, PV relocation is not allowed because the relocation of a PV in a Y CU changes the chroma components of its associated FOP pixel value, which should never be changed.

B. PV RELOCATION AND REACTIVATION

The reference range for PV is the same as that for BV and SV and is divided into a predetermined number of ***reference subrange***. The size of each reference subrange is *min* (64, LCU_size) × *min* (64, LCU_size), where LCU_size is the width and height of the largest coding unit (LCU). During the coding process, the old reference subrange is obsolete and removed from the reference range, and the new reference subrange is added to the reference range one by one. As shown in Figure 4, the reference subrange size is 64 × 64, and the reference range consists of four reference subranges, including the current reference subrange and three previous reference subranges, shown in gray. ***The current CU***(8 × 8) ***in Figure 4 is the same CU illustrated in the right part of Figure 1***.

PV relocation: Whenever a PV (and its associated PV address as well as FOP pixel value) is used by an Evs, if the PV points to a position in a previous reference subrange (i.e., not in the current reference subrange), then it is relocated to a new position in the Evs, that is, the value of the PV in PvInfoList is changed from the coordinates of the previous position to the coordinates of the new position. Because the new position will be removed from the reference range later than the previous position, the PV will be used by more EUSP CUs to improve the EUSP coding efficiency.

PV reactivation: As shown in Figure 4, a PV has three PV statuses: YUV-sleeping, UV-sleeping, and active. A YUV-sleeping PV is a PV pointing to a position in an obsolete reference subrange, which still exists in PvInfoList and HistPvInfoList. The YUV components of the associated FOP pixel value of a YUV-sleeping PV are temporarily stored in a sleeping PV buffer, denoted by SlpPVBuf. As shown in Figure 4, PV_0 and PV_1 are the two YUV-sleeping PVs at the start of coding an EUSP CU. An active PV is a PV that points to a position within a valid reference range. As shown in Figure 4, PV_2 and PV_3 in PvInfoList are active PVs. A UV-sleeping PV is a PV pointing to a new position in the valid reference range, which is in the middle of the transition from a YUV-sleeping PV to an active PV. If the new position is a top-left-2 × 2 position (top-left position of 2 × 2 pixels), its status immediately changes to active; otherwise, its status will change to active at a later time. The UV components of the associated FOP pixel value of a UV-sleeping PV are still temporarily stored in SlpPVBuf because the UV-sleeping PV points to a non-top-left-2 × 2 position which does not have room for the UV component. As shown in Figure4, PV_1 in PvInfoList is a UV-sleeping PV at the end of coding of an EUSP CU. The relation among the three PV statuses is as follows:

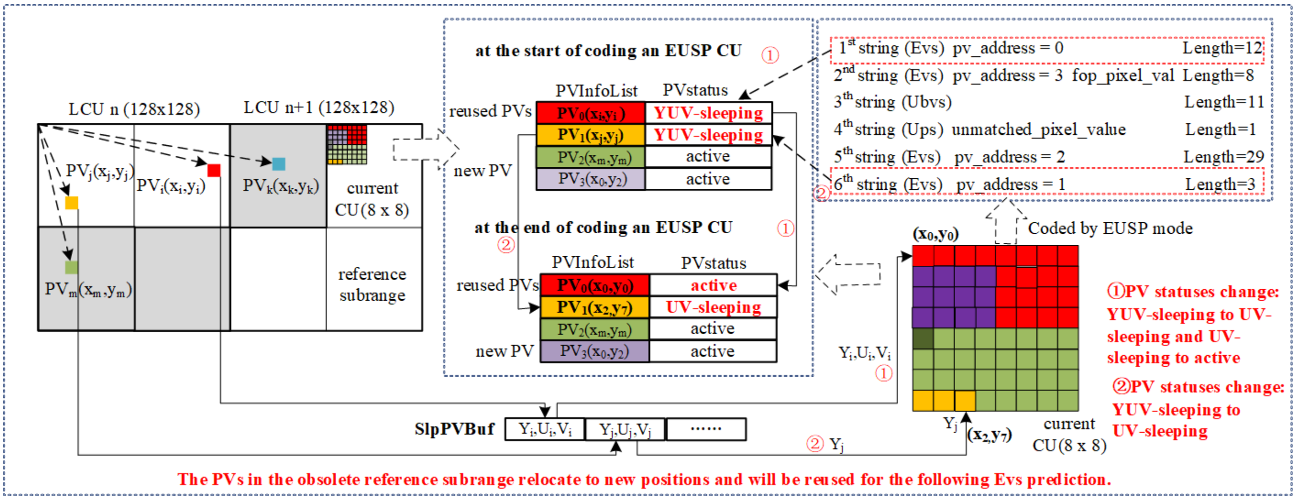


FIGURE 4. PV relocation and reactivation.

1) **From YUV-sleeping to UV-sleeping**: If a YUV-sleeping PV (and its associated PV address) reoccurs in a new position when it is used by an Evs in the current CU for the first time, the status of the PV is changed from YUV-sleeping to UV-sleeping, the value of the PV is replaced by the coordinates of the new position where the first pixel of the Evs is, and the Y component of its associated FOP pixel value stored in SlpPVBuf is put into the new position pointed to by the UV-sleeping PV. As shown in Figure 4, the 1st string in the current CU is an Evs using a YUV-sleeping PV with a value of (x_i, y_i); thus, (x_i, y_i) is replaced by (x_0, y_0), which is the coordinate of the first pixel of the 1st string, and the PV status is changed to UV-sleeping. The 6th string in the current CU is also an Evs using a YUV-sleeping PV with a value of (x_j, y_j); thus, (x_j, y_j) is replaced by (x_2, y_7), which is the coordinate of the first pixel of the 6th string, and the PV status is changed to UV-sleeping.
2) **From UV-sleeping to active:** If a UV-sleeping PV (and its associated PV address) is used by an Evs (including the 1st Evs, which changes the status of the PV from YUV-sleeping to UV-sleeping) or Ubvs with at least one top-left-2 × 2 position, the status of the PV is changed from UV-sleeping to active, the value of the PV is replaced by the coordinates of the first top-left-2 × 2 position, and the UV component of its associated FOP pixel value stored in SlpPVBuf is put into the position pointed to by the active PV. As shown in Figure 4, (x_0, y_0) is a top-left-2 × 2 position, thus, the status of the PV is changed to active immediately after it is changed to UV-sleeping. On the other hand, the UV-sleeping $PV_1(x_2, y_7)$ is not used by any Evs or Ubvs with any top-left-2 × 2 position in the current CU; thus, the status of the PV is still UV-sleeping at the end of coding the current CU and may be changed to active later in a future EUSP CU.
3) **From active to YUV-sleeping:** When a reference subrange is obsolete and removed from the reference range, the status of all PVs pointing to a position in the obsolete reference subrange is changed to YUV-sleeping.

C. LINE-BASED OVERLAPPING STRING PREDICTION

A line-based overlapping string is an Fms in the FPSP sub-mode or a Ubvs in the EUSP sub-mode with overlapping lines between the current string and reference string. Figure 5 illustrates an example of a line-based overlapping string prediction. Let CuWidth be the width of a CU, (SV_x, SV_y) be the SV of a string, and M be the height (i.e., the number of lines) of a string. A line-based overlapping string satisfies the following condition:

$$SV_y < 0 \text{ and } abs(SV_x) < \text{CuWidth and abs}(SV_y) < M \quad (1)$$

A line-based overlapping string can be split into N ($N \geq 2$) nonoverlapping strings, where N is calculated as follows:

$$N = \text{ceil}(M / abs(SV_y)) \quad (2)$$

M can be written as $M = (N\text{-}1)^* abs(SV_y) + n$, where $1 \leq n \leq \text{abs}(SV_y)$. Therefore, a line-based overlapping string with string height equal to M can be split into N-1 strings with string height equal to $abs(SV_y)$ and one string with string height equal to n. The resulting N strings have string heights equal to or smaller than $abs(SV_y)$ and the same vertical component SV_y of SV. Each N string is a nonoverlapping string because it has a string height equal to or smaller than abs (SV_y), which is the vertical distance between the current string and reference string. Function ceil (x) is the smallest integer greater than or equal to x. Function abs (x) returns the absolute value of x.

For example, as shown in Figure 5, the current string marked with the blue boundary has (SV_x, SV_y) equal to (0,

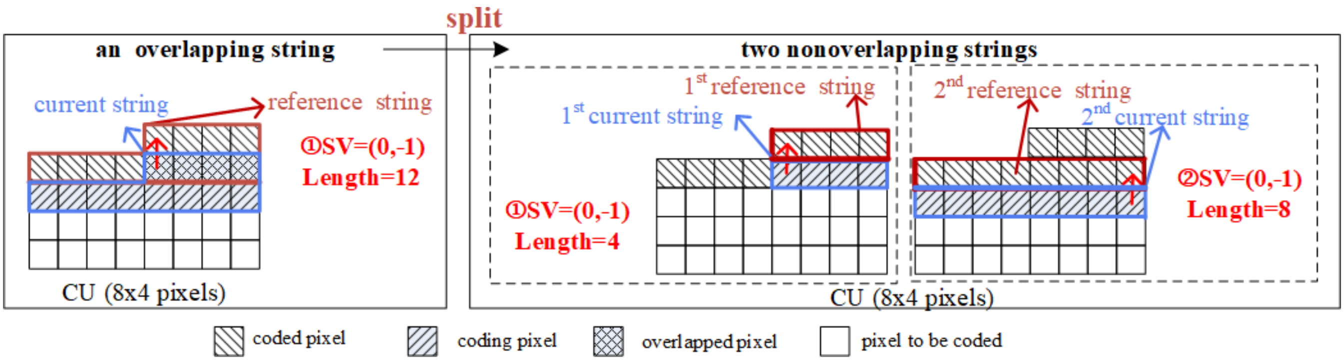


FIGURE 5. Line-based overlapping string prediction.

−1), length of 12, and string height of M equal to 2. It is a line-based overlapping string that can be split into two nonoverlapping strings with a string height of 1 ($abs\ (SV_y) = 1$).

There are two purposes for splitting a line-based overlapping string into multiple nonoverlapping strings, as described below.

1) Implementing an overlapping string copy with a nonoverlapping line copy has a much lower hardware implementation cost than nonoverlapping pixel copy.
2) Constraint on the total number of nonoverlapping strings plus unmatched pixels in a CU from exceeding a quarter of the total number of pixels in the CU reduces hardware implementation complexity and the cost of ISC.

D. OPTIMIZED CODING METHOD FOR STRING LENGTH

In the EUSP CU, a syntax element str_length_minus1 is used to code the string length for Evs, Ubvs, and Ups. The value range (denoted by C) of str_length_minus1 is [0, maxStrLength), that is, $0 \leq$ str_length_minus1 $<$ maxStrLength, where maxStrLength is the number of remaining pixels to be coded in the current CU. The value range of maxStrLength is [1, 1024].

It is observed that the value of str_length_minus1 has a unique statistical distribution. Based on the observation and minimum entropy principle, value range C is divided into multiple subranges for optimal coding and binarization. For the string lengths of Evs and Ups, there are eleven subranges, which are the intersection of C and [0,1), [1,2), [2,4), [4,8), [8,16), [16,32), [32,64), [64,128), [128,256), [256,512), and [512,1024), respectively, that is,

$$C \cap [0, 1),\ C \cap [2^k, 2^{k+1}),\ k = 0 \sim 9 \tag{3}$$

Clearly, $C \cap [2^k, 2^{k+1})$ is empty if maxStrLength $\leq 2^k$. Therefore, let K be the maximum integer satisfying $2^K <$ maxStrLength. Then, (3) can be rewritten as

$$C \cap [0, 1),\ C \cap [2^k, 2^{k+1}),\ k = 0 \sim K \tag{4}$$

TABLE 2. Fourteen AVS3-SCC CTC images used in the experiment.

Resolu-tion	Sequence name	Abbrevi-ation	Catego-ry	No. of frames
1920x 1080	sc_flyingGraphics	FLYG	TGM	600
	sc_desktop	DSK	TGM	600
	sc_console	CNS	TGM	600
	ChineseDocumentEditing	CDE	TGM	300
	EnglishDocumentEditing	EDE	TGM	300
	Spreadsheet	SPS	TGM	300
	BitstreamAnalyzer	BSA	TGM	300
	CircuitLayoutPresentation	CLP	TGM	300
	program	PRG	TGM	600
	web_en	WEBB	TGM	600
	word_excel	WDEC	TGM	600
	program_vidyo	PRGV	MC	600
	ArenaOfValor	AOV	A	600
	BQTerrace	BQT	CC	600

When maxStrLength > 2, (4) can also be rewritten as

$$C \cap [0, 1),\ C \cap [2^k, 2^{k+1}),\ C \cap [2^K, \text{maxStrLength}),\quad k = 0 \sim K - 1 \tag{5}$$

On the other hand, for the string length of Ubvs, there are ten subranges: the intersection of C and [0,2), [2,4), [4,8), [8,16), [16,32), [32,64), [64,128), [128,256), [256,512), and [512,1024), that is,

$$C \cap [0, 2),\ C \cap [2^k, 2^{k+1}),\ k = 1 \sim 9 \tag{6}$$

The syntax of str_length_minus1 consists of a prefix and suffix, which use truncated unary and truncated binary codes, respectively, for binarization.

V. EXPERIMENTS

A. EXPERIMENTAL SETTINGS

The experiments evaluate and compare the coding efficiency and complexity of the EISC using the following nine CODECs: The four proposed enhancement techniques, CU partition improvements, PV relocation and reactivation, line-based overlapping string prediction, and optimized coding method for string length are denoted as T1, T2, T3, and T4, respectively.

TABLE 3. Coding efficiency comparisons (BD-rate reduction %) among NINE codecs.

	Test sequences category	AI			LDB		
		Y	U	V	Y	U	V
HPM vs. HPM-T1Off	TGM (1080p)	-0.98	-0.24	-0.34	-0.78	0.05	-0.13
	MC (1080p)	-0.62	0.21	-0.09	-0.76	-0.04	0.04
	Average of TGM and MC	***-0.95***	***-0.20***	***-0.32***	***-0.78***	***0.04***	***-0.12***
HPM vs. HPM-T2Off	TGM (1080p)	-0.74	-0.73	-0.75	-0.45	-0.14	-0.36
	MC (1080p)	-0.70	-1.17	-1.18	-0.05	-0.45	0.13
	Average of TGM and MC	***-0.73***	***-0.77***	***-0.79***	***-0.42***	***-0.17***	***-0.32***
HPM vs. HPM-T3Off	TGM (1080p)	-0.51	-0.50	-0.53	-0.35	-0.12	-0.32
	MC (1080p)	-0.33	-0.30	-0.30	0.37	0.71	1.11
	Average of TGM and MC	***-0.50***	***-0.48***	***-0.51***	***-0.29***	***-0.05***	***-0.20***
HPM vs. HPM-T4Off	TGM (1080p)	-0.21	-0.16	-0.10	-0.17	-0.08	-0.01
	MC (1080p)	-0.33	-0.46	-0.58	-0.18	-0.94	-0.45
	Average of TGM and MC	***-0.22***	***-0.18***	***-0.14***	***-0.17***	***-0.01***	***-0.04***
HPM vs. HPM-EISCOff	TGM (1080p)	***-2.40***	***-1.69***	***-1.70***	***-1.48***	***-0.99***	***-0.96***
	MC (1080p)	***-2.33***	***-1.96***	***-1.97***	***-1.54***	***-1.25***	***-0.58***
	Average of TGM and MC	***-2.39***	***-1.71***	***-1.73***	***-1.49***	***-1.01***	***-0.93***
HPM vs. HPM-FPSPOff	TGM (1080p)	-4.64	-5.05	-5.23	-3.18	-3.24	-3.39
	MC (1080p)	-2.13	-2.49	-2.67	-1.31	-2.06	-0.26
	Average of TGM and MC	***-4.43***	***-4.84***	***-5.02***	***-3.03***	***-3.14***	***-3.13***
HPM vs. HPM-EUSPOff	TGM (1080p)	-4.27	-2.74	-2.65	-2.44	-1.11	-1.29
	MC (1080p)	-10.19	-9.63	-10.35	-5.07	-5.28	-6.31
	Average of TGM and MC	***-4.77***	***-3.31***	***-3.29***	***-2.66***	***-1.45***	***-1.71***
HPM vs. HPM-ISCOff	TGM (1080p)	-9.70	-8.46	-8.44	-5.98	-4.91	-5.08
	MC (1080p)	-14.54	-14.76	-15.28	-7.74	-6.71	-7.34
	Average of TGM and MC	***-10.10***	***-8.98***	***-9.01***	***-6.13***	***-5.06***	***-5.26***

1) HPM, the latest AVS3 reference software [30].
2) HPM-T1Off, T1 off in HPM.
3) HPM-T2Off, T2 off in HPM.
4) HPM-T3Off, T3 off in HPM.
5) HPM-T4Off, T4 off in HPM.
6) HPM-EISCOff, T1, T2, T3, and T4 off in HPM.
7) HPM-EUSPOff, EUSP sub-mode off in HPM.
8) HPM-FPSPOff, FPSP sub-mode off in HPM.
9) HPM-ISCOff, FPSP and EUSP sub-mode off in HPM.

All experimental results are generated under the CTC and configurations defined in [31] for AVS3 SCC. The BD-rate metric [32] is used to evaluate the overall coding efficiency improvement. Fourteen YUV 4:2:0 test sequences, shown in Table 2, classified into text and graphics with motion (TGM), mixed content (MC), gaming (G), and camera captured (CC) categories, are used in the experiment. For each category, the average BD rate reduction was calculated separately for the Y, U, and V color components. Four QPs (27, 32, 38, and 45) and two configurations of all intra (AI) and low-delay B (LDB) are tested.

To evaluate the encoder and decoder complexity, the encoding and decoding software runtimes are also compared. The runtime measurement is not accurate and may have a measurement error of a few percent.

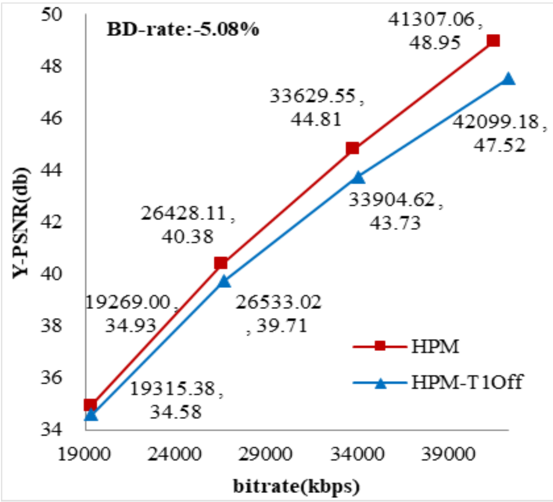
B. EXPERIMENTAL RESULTS AND CONCLUSION

To evaluate the coding efficiency of each ISC enhancement technique, overall EISC, and ISC, the coding efficiency comparisons between HPM and other codecs for the AI and LDB configurations are shown in Table 3. Because the ISC does not change the coding efficiency for the G and CC categories, the experimental results for the G and CC categories are not shown in Table 3.

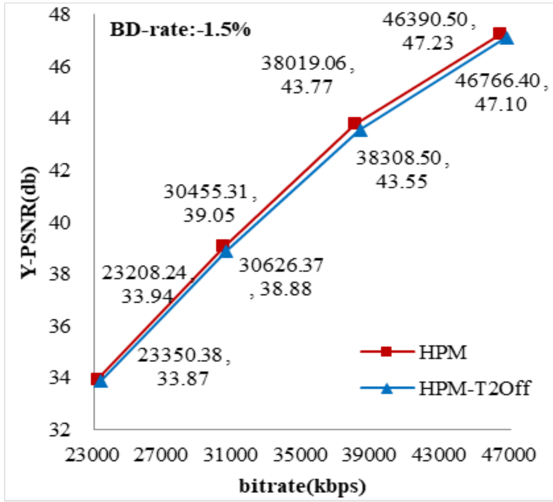
The RD curves and complexity comparisons among the six codecs (HPM, HPM-T1Off, HPM-T2Off, HPM-T3Off, HPM-T4Off, and HPM-EISCOff) for the AI configuration are shown in Figure 6. The encoding and decoding runtime ratios among the HPM and other codecs are listed in Table 4.

The experimental results can be summarized as follows.

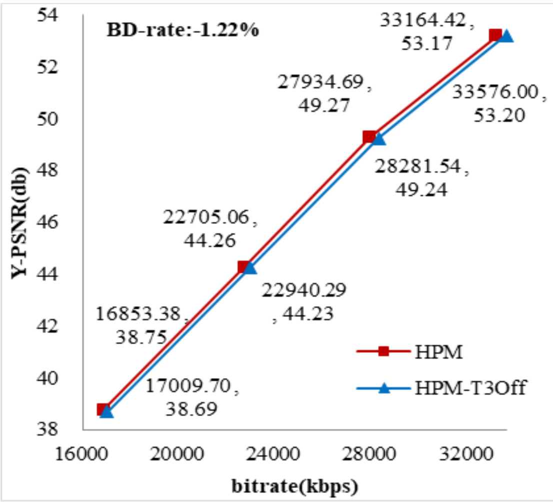
1) Each proposed ISC enhancement technique improves the ISC coding efficiency. As shown in Table 3, for the 12 TGM and MC sequences, T1 achieves an average Y BD-rate reduction ***0.95%*** for AI and ***0.78%*** for LDB, T2 achieves an average Y BD-rate reduction



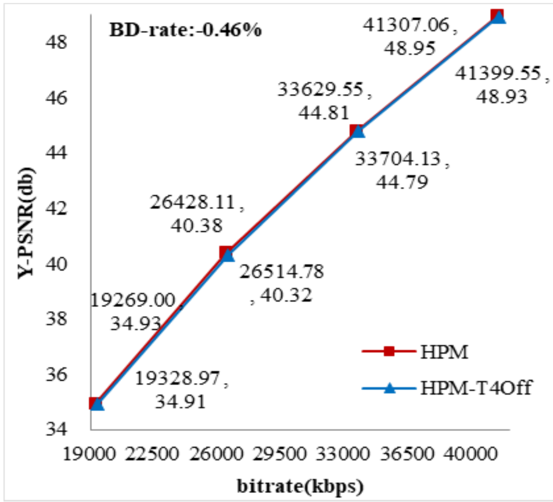
a) RD curves of CLP (HPM vs. HPM-T1Off).



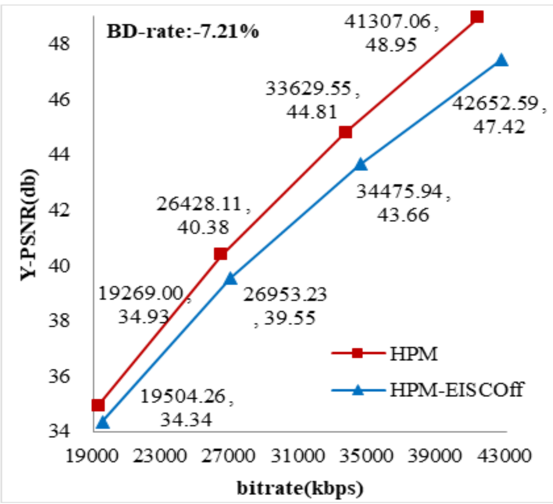
b) RD curves of BSA (HPM vs. HPM-T2Off).



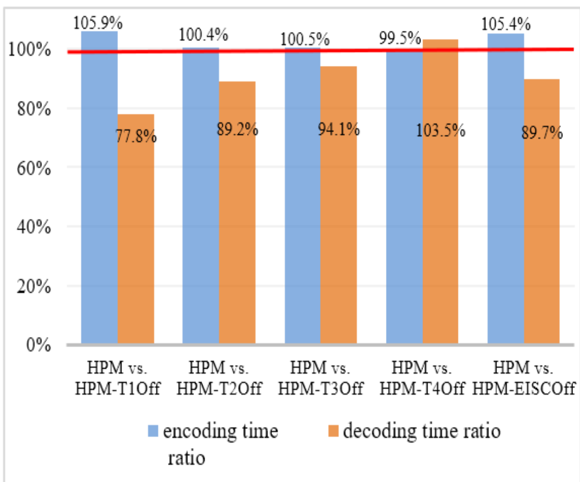
c) RD curves of CNS (HPM vs. HPM-T3Off).



d) RD curves of CLP (HPM vs. HPM-T4Off).



e) RD curves of BSA (HPM VS. HPM- EISCOff).



f) Complexity among six codecs.

FIGURE 6. RD curves and complexity comparisons among six codecs for AI configuration.

TABLE 4. Encoding and decoding runtime COMPARISONs among NINE codecs.

	encoding time ratio [%]		decoding time ratio [%]	
	AI	LDB	AI	LDB
HPM vs. HPM-T1Off	105.92	100.57	77.81	84.57
HPM vs. HPM-T2Off	100.42	103.01	89.17	85.72
HPM vs. HPM-T3Off	100.47	102.43	94.06	98.63
HPM vs. HPM-T4Off	99.48	99.24	103.45	101.80
HPM vs. HPM-EISCOff	***105.36***	***104.49***	***89.72***	***85.17***
HPM vs. HPM-FPSPOff	110.53	105.39	96.41	99.32
HPM vs. HPM-EUSPOff	107.12	104.42	96.71	102.23
HPM vs. HPM-ISCOff	***115.87***	***106.89***	***98.80***	***92.67***

0.73% for AI and ***0.42***% for LDB, T3 achieves an average Y BD-rate reduction ***0.50***% for AI and ***0.29***% for LDB, and T4 achieves an average Y BD-rate reduction ***0.22***% for AI and ***0.17***% for LDB. As shown in Figure 6, T1, T2, T3, and T4 can achieve a Y BD-rate reduction of ***5.08%*** for CLP, ***1.5%*** for BSA, ***1.22%*** for CNS, ***0.46%*** for CLP.

2) The proposed ISC enhancement techniques achieve a notable overall coding efficiency improvement. As shown in Table 3, for 12 TGM and MC sequences, EISC can achieve an average Y BD-rate reduction ***2.39***% for AI and ***1.49***% for LDB. For the TGM category, EISC achieves average Y, U, and V BD-rate reductions of ***2.40***%, ***1.69***%, ***1.70***% for AI, and ***1.48***%, ***0.99***%, ***0.96***% for LDB, respectively. For the MC category, EISC achieves average Y, U, and V BD-rate reductions of ***2.33***%, ***1.96***%, ***1.97***% for AI, and ***1.54***%, ***1.25***%, ***0.58***% for LDB, respectively. As shown in Figure 6, the EISC achieves a Y BD-rate reduction of ***7.21%*** for BSA.
3) With EISC on, the FPSP sub-mode, EUSP sub-mode, and ISC mode notably improve the SCC coding efficiency. As shown in Table 3, for the 12 TGM and MC sequences, the FPSP sub-mode can achieve an average Y, U, and V BD-rate reduction of ***4.4***%, ***4.9***%, ***5.0***% for AI, and ***3.0***%, ***3.1***%, ***3.1***% for LDB, respectively. The EUSP sub-mode can achieve an average Y, U, and V BD-rate reduction of ***4.8***%, ***3.3***%, ***3.3***% for AI, and ***2.7***%, ***1.5***%, ***1.7***% for LDB. The ISC mode can achieve an average Y, U, and V BD-rate reduction of ***10.1***%, ***9.0***%, ***9.0***% for AI, and ***6.1***%, ***5.1***%, ***5.3***% for LDB, respectively.
4) Compared with the HPM, the overall encoding runtime increase of the EISC is minor. As shown in Table 4, the encoding time increase of EISC was ***5.36***% for AI and ***4.49***% for LDB. The increase mostly comes from T1, which extends the EUSP sub-mode to Y CUs and small CUs with a width and/or height equal to four pixels. The encoding times of T2, T3, and T4 are almost the same as those of the HPM. Table 4 also shows that the decoding runtimes of the T1, T2, T3, T4, EISC, FPSP sub-mode, EUSP sub-mode, and ISC mode are less than or almost the same as those of the HPM. The reason for the reduction in decoding runtime is that the decoding operation of the ISC is nothing but simply copying strings.

VI. CONCLUSION AND FUTURE WORK

An efficient SCC tool called intra string copy (ISC) has been adopted in AVS3. To further improve the coding efficiency of SCC, this paper proposes four enhancement techniques of ISC (EISC), including ISC-mode CU partition improvements, PV relocation and reactivation, line-based overlapping string prediction, and an optimized coding method for string length in the EUSP sub-mode. The EISC coding system architecture is illustrated, and the technical details of the EISC are elaborated. The experimental results show that EISC can improve coding efficiency with low additional encoding and almost no additional decoding complexity.

Block-based video coding has been studied for half a century, whereas string-based video coding is still in its early stages. Future work includes 1) applying ISC to the next-generation video coding standard; 2) improving the coding performance of ISC in specific application scenarios [34] [35], especially in mobile wireless network application scenarios; 3) reducing the coding complexity with negligible coding performance [36], [37] of ISC; 4) applying deep learning and neural network [38] approaches to video analysis and ISC-based video coding.

REFERENCES

[1] X. Xu and S. Liu, "Overview of screen content coding in recently developed video coding standards," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 2, pp. 839–852, Feb. 2022.
[2] T. Nguyen, X. Xu, F. Henry, R.-L. Liao, M. G. Sarwer, M. Karczewicz, Y.-H. Chao, J. Xu, S. Liu, D. Marpe, and G. J. Sullivan, "Overview of the screen content support in VVC: Applications, coding tools, and performance," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3801–3817, Oct. 2021.
[3] S. Ma, L. Zhang, and S. Wang, "Evolution of AVS video coding standards: Twenty years of innovation and development," *Sci. China Inf. Sci.*, to be published.
[4] J. Zhang, C. Jia, M. Lei, S. Wang, S. Ma, and W. Gao, "Recent development of AVS video coding standard: AVS3," in *Proc. Picture Coding Symp. (PCS)*, Ningbo, China, Nov. 2019, pp. 1–5.
[5] A. V. Group, *Minutes of the 74th Plus AVS Video Group Meeting*, Audio Video Coding Standard Team, Teleconference, China, Standard AVS-N2955, Oct. 2020.
[6] Y. Wang, X. Xu, and S. Liu, "Intra block copy in AVS3 video coding standard," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, London, U.K., Jul. 2020, pp. 1–6.
[7] J. Li, L. Zhang, and K. Zhang, *Frequency-Based Intra Mode Coding*, Audio Video Coding Standard Team, Haikou, China, Standard AVS-M4972, Aug. 2019.
[8] Y. Zhang, K. Zhang, L. Zhang, H. Liu, Y. Wang, S. Ma, and W. Gao, "Implicit seleted transform skip method for AVS3," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Anchorage, AK, USA, Sep. 2021, pp. 2134–2138.
[9] Y. Wang, X. Xu, and S. Liu,, *Deblocking Switch Control for Screen Content Encoding*, Audio Video Coding Standard Team, Teleconference, China, Standard AVS-M5221, Apr. 2020.

[10] Y. Wang, X. Xu, and S. Liu, "Low complexity implementation of intra string copy in AVS3," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Shenzhen, China, Jul. 2021, pp. 1–4.
[11] Q. Zhou, L. Zhao, K. Zhou, T. Lin, H. Wang, S. Wang, and M. Jiao, "String prediction for 4:2:0 format screen content coding and its implementation in AVS3," *IEEE Trans. Multimedia*, vol. 23, pp. 3867–3876, 2021.
[12] K. Zhou, L. Zhao, Y. Zigao, H. Wang, T. Lin, F. Sheng, and Y. Yufen, "Equal value string and copy above string based string prediction for SCC in AVS3," *IEEE Trans. Multimedia*, early access, Nov. 19, 2021, doi: 10.1109/TMM.2021.3129358.
[13] L. Zhao, K. Zhou, T. Lin, and J. Guo, "A universal string prediction approach and its application in AVS2 mixed content coding," *Chin. J. Comput.*, vol. 42, no. 41, pp. 1–15, Apr. 2019.
[14] T. Lin, P. Zhang, S. Wang, K. Zhou, and X. Chen, "Mixed chroma sampling-rate high efficiency video coding for full-chroma screen content," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 1, pp. 173–185, Jan. 2013.
[15] W. Zhu, J. Xu, W. Ding, Y. Shi, and B. Yin, "Adaptive LZMA-based coding for screen content," in *Proc. Picture Coding Symp. (PCS)*, 2013, San Jose, CA, USA, Dec. 2013, pp. 373–376.
[16] B. Li, J. Xu, and F. Wu, "1-D dictionary mode for screen content coding," in *Proc. IEEE Int. Conf. Vis. Commun. Image Process.*, Valletta, Malta, Dec. 2014, pp. 189–192.
[17] F. Zou, Y. Chen, M. Karczewicz, and V. Seregin, "Hash based intra string copy for HEVC based screen content coding," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Turin, Italy, Jun. 2015, pp. 1–4.
[18] L. Zhao, T. Lin, K. Zhou, S. Wang, and X. Chen, "Pseudo 2D string matching technique for high efficiency screen content coding," *IEEE Trans. Multimedia*, vol. 18, no. 3, pp. 339–350, Mar. 2016.
[19] L. Zhao, T. Lin, and K. Zhou, "An efficient ISC offset parameter coding algorithm in screen content coding," *Chin. J. Comput.*, vol. 40, no. 5, pp. 1–11, May 2017.
[20] Z. Ma, W. Wang, M. Xu, and H. Yu, "Advanced screen content coding using color table and index map," *IEEE Trans. Image Process.*, vol. 23, no. 10, pp. 4399–4412, Oct. 2014.
[21] L. Zhao, K. Zhou, J. Guo, S. Wang, and T. Lin, "A universal string matching approach to screen content coding," *IEEE Trans. Multimedia*, vol. 20, no. 4, pp. 796–809, Apr. 2018.
[22] K. Zhou, L. Zhao, and T. Lin, "A flexible and uniform string matching technique for general screen content coding," *Multimedia Tools Appl.*, vol. 77, no. 18, pp. 23751–23775, Sep. 2018.
[23] L. Zhao, K. Zhou, and J. Guo, "Pixel string matching for full-chroma screen and mixed content coding in AVS2," *Chin. J. Comput.*, vol. 41, no. 11, pp. 2482–2495, Oct. 2018.
[24] L. Zhao, T. Lin, and J. Guo, "Universal string prediction-based inter coding algorithm optimization in AVS2 mixed content coding," *Chin. J. Comput.*, vol. 42, no. 28, pp. 1–13, Mar. 2019.
[25] Y. Wang, T. Lin, and L. Zhao, *CE_SCC.1 Related: Storage Optimization for Equal Value String Reference Pixel*, Audio Video Coding Standard Team, Teleconference, China, Standard AVS-M5822, Oct. 2020.
[26] Q. Zhou, L. Zhao, and H. Wang, *CE_SCC_Related: Encoding and Decoding Optimization for String Vector Based on the Inherent Characteristics of Several Above Strings*, Audio Video Coding Standard Team, Teleconference, China, Standard AVS-M5950, Dec. 2020.
[27] Y. Wang, Q. Zhou, and L. Zhao, *SCC: Unit Basis Vector String for String Prediction*, Audio Video Coding Standard Team, Teleconference, China, Standard AVS-M5994, Dec. 2020.
[28] H. Wang, L. Zhao, and Z. Ye, *Extending to CUs With a Width or Height of 4 and TreeL Component Mode for the Evs-Ubvs Sub-Mode*, Audio Video Coding Standard Team, Teleconference, China, Standard AVS-M6256, Feb. 2020.
[29] W. Zhang, L. Zhao, and Y. Yang, *Optimization of String Length Coding for Equal Value String and Unit Basis Vector String*, Audio Video Coding Standard Team, Teleconference, China, Standard AVS-M6108, Feb. 2021.
[30] *AVS3 Reference Software*. Accessed: May 27, 2021. [Online]. Available: https://gitlab.com/AVS3_Software/hpm/-/tree/HPM-11.1
[31] X. Xu, *AVS3-P2 Common Test Conditions for Screen Content Coding*, Audio Video Coding Standard Team, Teleconference, China, Standard AVS-M5928, Sep. 2020.
[32] G. Bjøntegaard, *Calculation of Average PSNR Differences Between RD-100 Curves*, Document VCEG-M33, ITU-T SG16 Q.6, Austin, Texas, Apr. 2001.
[33] C. Sun, J. Li, and J. Xu, *IBC Virtual Buffer*, Audio Video Coding Standard Team, Shenzhen, China, Standard AVS-M5097, Dec. 2019.
[34] Y. Yang, K. Zhou, L. Zhao, and T. Lin, "An ultralow complexity string matching approach to screen content coding in AVS3," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 9, pp. 3714–3718, Sep. 2021.
[35] L. Zhao, T. Lin, D. Zhang, K. Zhou, and S. Wang, "An ultra-low complexity and high efficiency approach for lossless alpha channel coding," *IEEE Trans. Multimedia*, vol. 22, no. 3, pp. 786–794, Mar. 2020.
[36] B. Heidari and M. Ramezanpour, "Reduction of intra-coding time for HEVC based on temporary direction map," *J. Real-Time Image Process.*, vol. 17, no. 3, pp. 567–579, Jun. 2020.
[37] N. Najafabadi and M. Ramezanpour, "Mass center direction-based decision method for intraprediction in HEVC standard," *J. Real-Time Image Process.*, vol. 17, no. 5, pp. 1153–1168, Oct. 2020.
[38] S. Ma, X. Zhang, C. Jia, Z. Zhao, S. Wang, and S. Wang, "Image and video compression with neural networks: A review," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 6, pp. 1683–1698, Jun. 2020.

LIPING ZHAO (Member, IEEE) received the B.S. degree in computer science and technology from Hengyang Normal University, in 2006, the M.S. degree in computer science and technology from Hunan University, in 2009, and the Ph.D. degree from Tongji University, Shanghai, China, in 2017.

She was with the College of Mathematics, Physics and Information Engineering, Jiaxing University, from July 2009 to June 2017. She is currently an Associate Professor with the Department of Computer Science and Engineering, Shaoxing University, Shaoxing, China. She is also a Researcher with the Information Technology Research and Development Innovation Center, Peking University, Shaoxing. Her current research interest includes image or video coding.

HUIHUI WANG received the B.S. degree in information engineering from the East China University of Science and Technology, in 2019. She is currently pursuing the M.E. degree with Tongji University, Shanghai. Her current research interest includes screen content coding.

QINGYANG ZHOU received the B.E. degree in communication engineering from Shanghai University, in 2018, and the M.E. degree in control science and engineering from Tongji University, in 2021.

His current research interest includes screen content coding.

WENJUAN ZHANG received the B.E. degree in electronic science and technology from Anhui University, in 2019. She is currently pursuing the M.E. degree with Tongji University, Shanghai.

Her current research interest includes screen content coding.

KAILUN ZHOU received the B.S. and M.S. degrees from Shanghai Jiaotong University, Shanghai, China, in 2000 and 2003, respectively, and the Ph.D. degree from Tongji University, Shanghai, in 2017.

His current research interests include embedded system design and video coding.

KELI HU (Member, IEEE) received the Ph.D. degree in communication and information systems from the Shanghai Institute of Microsystem and Information Technology, University of Chinese Academic and Sciences, Shanghai, China, in 2014. He is currently an Associate Professor with the Department of Computer Science and Engineering, Shaoxing University, Shaoxing, China.

His current research interests include machine learning and image processing.

TAO LIN received the B.S. degree from East China Normal University, Shanghai, China, in 1982, and the M.S. and Ph.D. degrees from Tohoku University, Sendai, Japan, in 1985 and 1989, respectively. Since 2003, he has been with the VLSI Laboratory, Tongji University, Shanghai. In 2005, he was awarded "Changjiang Scholars," the highest honor given by China Ministry of Education. From 1988 to 2002, he was with University of California at Berkeley as a Postdoctoral Researcher, IDT, PMC-Sierra, Cypress Semiconductor, and NeoMagic, all in Silicon Valley. His current research interests include video coding and SoC design.