IEICE TRANSACTIONS

on Information and Systems

DOI:10.1587/transinf.2024EDP7056

Publicized:2024/09/24

A PUBLICATION OF THE INFORMATION AND SYSTEMS SOCIETY
The Institute of Electronics, Information and Communication Engineers
Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3 chome, Minato-ku, TOKYO, 105-0011 JAPAN

PAPER

Adaptive Merge Candidate Selection based on Geometric Partitioning Mode beyond Versatile Video Coding

Haruhisa KATO[†a)], *Senior Member*, **Yoshitaka KIDANI**[†], *Member*, *and* **Kei KAWAMURA**[†], *Senior Member*

SUMMARY We propose a method for adaptively selecting merge candidates for the geometric partitioning mode (GPM) in versatile video coding (VVC). The conventional GPM contributes to improved coding efficiency and subjective quality by partitioning the block into two nonrectangular partitions with motion vectors. The motion vector of the GPM is encoded as an index of the merge candidate list, but it does not consider that the GPM partitions are nonrectangular. In this paper, the distribution of merge candidates was evaluated for each GPM mode and partition, and a characteristic bias was revealed. To improve the coding efficiency of VVC, the proposed method allows GPM to select merge candidates that are specific to the partition. This method also introduces adaptive reference frame selection using template matching of adjacent samples. Following common test conditions in the Joint Video Experts Team (JVET), the experimental results showed an improvement in coding efficiency, with a bitrate savings of 0.16%, compared to the reference software for exploration experiments on enhanced compression beyond VVC capability in the JVET.

key words: *video compression, versatile video coding (VVC), geometric partitioning mode (GPM), inter prediction, merge candidate list*

1. Introduction

Video coding has become an essential component of modern communication systems. Versatile video coding (VVC) has emerged as an international standard in this field [1]. The VVC standard, also known as H.266, is a successor to high-efficiency video coding (HEVC), also known as the H.265 standard [2]. They were developed by the Joint Video Experts Team (JVET), a collaborative team of the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG). VVC can reduce the size of video data by up to 50% with the same subjective video quality compared to HEVC [3].

A key feature of VVC is the geometric partitioning mode (GPM), which provides a new inter prediction for partitioning video frames [4]. Unlike conventional partitioning methods that divide frames into rectangular blocks, GPM allows for triangular and trapezoidal shapes, enabling a more precise video content representation. This flexibility in partitioning is particularly beneficial for complex video scenes that may be divided diagonally, which conventional block partitioning cannot handle.

Merge modes have been used for motion vector coding since HEVC and are also employed as a crucial component in GPM. The merge mode encodes the index of the merge candidate list, which is a list of potential reference blocks represented by their associated motion vectors that can be used for inter prediction [5]. Several merge candidates are listed from the motion vectors of adjacent blocks. The encoder can select the most suitable reference block by constructing a list of such candidates and encoding only the list index. In the GPM merge mode, each partition of the GPM has an index for selecting a motion vector from the merge candidate list. Although the merge candidate index reduces the coding of motion vectors, the current method of constructing the merge candidate list in the GPM does not adapt to the shape characteristics of the GPM. This approach may not always yield the optimal list construction for different types of shapes in GPM, especially those with complex geometric structures or textures. Therefore, there is room for improvement in coding efficiency because GPM uses a merge candidate index that ignores nonrectangular partitioning.

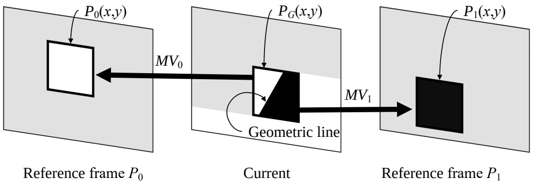


Fig. 1 The geometric partitioning mode (GPM) and reference frames are depicted in the image. The gray area represents the encoded region, while the white and black areas represent the reference samples. The GPM is utilized to predict the current block.

Our study proposes an adaptive method for constructing the merge candidate list based on a preliminary analysis of the relationships between the merge candidate list and the GPM modes. Specifically, we propose a method to adaptively select motion vectors as merge candidates only from blocks adjacent to each partition. Our findings can provide valuable insights for future research and development in video coding technology.

The remainder of this paper is organized as follows. Section 2 describes the related work in the field and highlights the challenges and gaps that our study aims to address. In Section 3, the relationships between GPM merge candidates and the selection rate of each partition are analyzed via a preliminary experiment. Section 4 presents a detailed description of our proposed method

[†] KDDI Research, Inc., 2-1-15, Ohara, Fujimino-shi, Saitama, 356-8502 JAPAN
a) E-mail: ha-katou@kddi.com

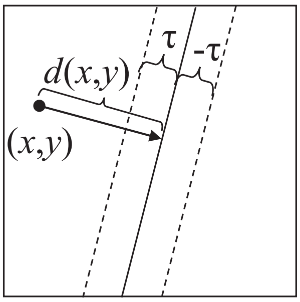
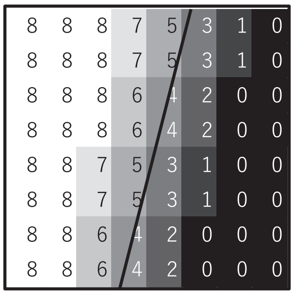


Fig. 2 GPM blending area around the geometric line between the dashed lines defined by width τ.



8	8	8	7	5	3	1	0
8	8	8	7	5	3	1	0
8	8	8	6	4	2	0	0
8	8	8	6	4	2	0	0
8	8	7	5	3	1	0	0
8	8	7	5	3	1	0	0
8	8	6	4	2	0	0	0
8	8	6	4	2	0	0	0

Fig. 3 Example of the weight coefficient W.

for adaptive merge candidate selection based on GPM partitioning. Section 5 shows the experimental results compared to those of the conventional method. Finally, in Section 6, we conclude the paper by summarizing our findings and suggesting directions for future research.

2. Related work

2.1 Geometric partitioning mode (GPM)

The GPM is a key tool within the VVC that was designed to enhance inter prediction [6]. In the context of GPM, the reference lists, denoted as L_0 and L_1, play a crucial role. These lists contain references to other frames in the video sequence that are used to predict the current frame. The specific reference frames used for prediction are labeled P_0 and P_1. These frames are selected from the lists L_0 and L_1, respectively, based on criteria such as prediction accuracy. The motion vectors MV_0 and MV_1 are essential to the inter prediction process. A motion vector is a two-dimensional vector that represents the displacement of a point from its position in a reference frame (P_0 or P_1) to its current block in the current frame. MV_0 corresponds to the motion vector derived from the reference frame P_0, while MV_1 corresponds to the motion vector derived from the reference frame P_1. In summary, the GPM in VVC utilizes the concepts of reference lists (L_0 and L_1), specific reference frames (P_0 and P_1), and motion vectors (MV_0 and MV_1) to perform highly efficient inter prediction.

The GPM allows the coding unit (CU) to be partitioned diagonally into two partitions along a geometric line. The position and angle of the line are determined by the GPM parameters, which are explicitly signaled within the bitstream. The GPM can support 64 different partitioning modes, providing high flexibility in partitioning the CU. The GPM is particularly effective for video sequences that contain geometric shapes or structures, such as diagonal edges. The GPM can handle complex textures and shapes more efficiently than can conventional partitioning modes, improving both objective and subjective quality.

Fig. 1 shows the prediction process of the GPM, wherein the left and right partitions of the coding blocks represent the MV_0 and MV_1 predictions of the reference frames P_0 and P_1, respectively. The predicted samples $P_G(x, y)$ within the GPM block are generated from two sets of motion-compensated samples $P_0(x, y)$ and $P_1(x, y)$ and weight coefficients $W(x, y)$ in Eq. (1).

$$\begin{aligned} P_G(x, y) =& (W(x, y) \times P_0(x, y) \\ &+ (8 - W(x, y)) \times P_1(x, y))/8, \end{aligned} \tag{1}$$

where the weight coefficient $W(x, y)$ in Eq. (2) for the coordinates (x, y) in the block ranges from 0 to 8 depending on the displacement $d(x, y)$ from the geometric line.

$$W(x, y) = \begin{cases} 0 & d(x, y) \leq -\tau \\ \frac{8}{2\tau}(d(x, y) + \tau) & -\tau < d(x, y) < \tau \ , \\ 8 & d(x, y) \geq \tau \end{cases} \tag{2}$$

where the parameter τ is the width of the blending area around the geometric line. Fig. 2 shows the displacement $d(x, y)$ and the blending area defined by τ around the geometric line. Fig. 3 shows an example of the weight coefficient $W(x, y)$. The values 0 and 8 remain unblended, and motion-compensated samples are directly employed. Values ranging from 1 to 7 are blended according to the displacement $d(x, y)$ using motion-compensated samples.

The following methods have been proposed to improve GPM coding efficiency further. Meng et al. proposed a method to determine the most probable GPM mode by evaluating the correlation between the coding block and its adjacent blocks [7]. The method achieves a bitrate saving of 0.17% relative to the GPM of the reference software, the VVC Test Model (VTM) 6. Additionally, they proposed a method for searching for the reference block of GPM using template matching [8]. This method improved the bitrate savings to 0.23% compared to GPM for VTM 8. The encoding and decoding times are 102% and 106%, respectively, with a particularly large increase in decoding time due to template matching. These methods use edge detection and the continuity of adjacent GPM modes to reduce the side information of the GPM. Kidani et al. proposed a method to allow inter and intra prediction mode for each GPM partition [9]. The method also proposed restricting the applicable intra prediction mode to optimize the coding efficiency. The method achieves

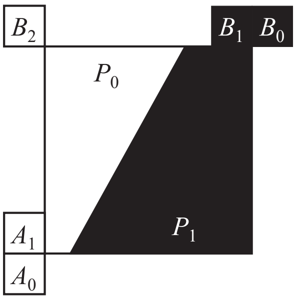


Fig. 4 Five adjacent spatial locations for the merge candidates (the left block A_1, the bottom-left block A_0, the above blocks B_1, the above-right block B_0 and the above-left block B_2).

a bitrate saving of 0.23% compared to the GPM in VTM 11 with an increase in encoding time to 107%. Lee et al. proposed a method to select two types of weight coefficients predicted on the edginess analysis of the GPM geometric line without changing the syntax modification [10]. A 0.21% bitrate savings were realized for screen content relative to the GPM of VTM 12. Nonetheless, the results for the overall camera-captured content remain unclear. Kato et al. proposed a method to select multiple boundary widths within the GPM via blockwise signaling [11–13]. It achieves a bitrate saving of 0.11% without increasing encoding and decoding times compared to GPM in VTM 15. Deng et al. proposed signaling difference vectors that compensate for the GPM motion vector [14]. The difference vectors are limited to eight directions and nine different distances. A gain of 0.27% was obtained in 105% of the encoding time compared to the enhanced compression model (ECM) 1.0. ECM is a reference software for activities beyond VVC that was developed by JVET. ECM 4.0 [15] shows a coding gain of approximately 16% compared to VTM 11 [16], which makes it more difficult for coding tools to bring additional gains. Chen et al. proposed restricting the GPM mode with template matching [17]. This method calculates the template matching cost between adjacent samples and the sample to which the motion vector refers by extending 64 modes of GPM geometric lines and restricting the GPM mode to a smaller cost. A 0.21% improvement in coding efficiency was obtained at 102% encoding time compared to that of ECM 3.1.

In summary, with the adoption of more sophisticated coding tools, gains are more challenging to achieve with newer reference software, but gains of 0.11% to 0.27% with a 100% to 107% increase in encoding time are typical performances for conventional methods of improving GPM. Although these methods refine GPM, they do not consider the impact of GPM partitioning on encoding motion vectors.

2.2 Spatial merge candidates in GPM

The motion vector for each partition within the GPM is coded utilizing the spatial merge mode. The spatial merge mode constructs a merge candidate list for each partition using the motion vectors of adjacent blocks as merge candidates. The list consists of merge candidates, and the samples indicated by the motion vectors of the reference frame P_0 or P_1 are used as references for inter prediction. The encoder selects the better merge candidate for each partition from the merge candidate list. The selection is based on the prediction error, which is the difference between the original and the predicted blocks. The encoder aims to minimize the prediction error to achieve high coding efficiency. Once the better merge candidate is selected, its associated index of the list is encoded into the bitstream instead of directly encoding the motion vectors. Here, the smaller the index is, the smaller the code volume, so it is directly related to improving coding efficiency to construct the list so that the better merge candidate is at the top of the list.

Table 1 Motion vectors used for the merge candidate list in GPM [18].

Merge candidate index	L_0 motion vector	L_1 motion vector
0	x	
1		x
2	x	
3		x
4	x	
5		x

In summary, encoding motion vectors in GPM using the merge candidate list is a key process in achieving efficient video compression in the VVC standard. For example, as shown in Fig. 4, in the spatial merge mode, the merge candidate list is constructed by listing the motion vectors of available adjacent blocks (the left block A_1, the bottom-left block A_0, the above blocks B_1, the above-right block B_0 and the above-left block B_2).

However, the conventional method of constructing the merge candidate list does not consider GPM partitions. For example, partition $P_1(x, y)$ in Fig. 4 has a different motion than partition $P_0(x, y)$, so the motion vector from the left block (A_1) is less likely to be used than the above block (B_1). There is a potential problem in constructing a list from blocks that are not adjacent to partitions, degrading coding efficiency.

Moreover, the conventional GPM has problems determining the list L_0 or L_1. While the merge candidate list is constructed with a set of motion vectors with lists L_0 and L_1, a coding block is constrained to a maximum of two motion vectors, and each GPM partition must select either L_0 or L_1 motion vectors. However, since there is no signal to specify the list, it is impossible to determine either L_0 or L_1. The conventional GPM assigns L_0 when the merge candidate index is even and L_1 when the merge candidate index is odd [18]. In Table 1, corresponding to each merge candidate index, the motion vector denoted by "x" is selected for the GPM. The conventional method has the problem that the number of merge candidates is halved. For example, if L_1 with index 0 is optimal, it cannot be selected. It is necessary to consider an adaptive method of selecting L_0 and L_1 without signaling to improve coding efficiency.

3. The distribution of merge candidates for each GPM partition

The coding efficiency of the GPM is more significant when a singular block contains different partitions, such as when a dynamic foreground intersects a static background. In this case, each partition tends to select motion vectors that belong to the same object in adjacent blocks. However, the conventional GPM follows a merge candidate list that assumes a rectangular shape and does not consider a partitioned nonrectangular shape.

To understand the impact of the GPM's lack of consideration of nonrectangular shapes on coding efficiency, we examined the number of merge candidates selected for each partition. Fig. 5 through Fig. 9 show the number of merge candidates in the five GPM partition modes (0, 12, 18, 27, 55) among the total of 64 modes. Since the coding efficiency of GPM depends on the resolution of the test sequence [8], we evaluated the GPM in high-resolution and low-resolution sequences using ECM 4.0 [15, 19].

Fig. 5 shows the results for GPM mode 0, where partition P_0 is adjacent to the above and left blocks and partition P_1 is adjacent only to the above block within Class A test sequences whose resolution is 3840 × 2160. Partition P_0 tends to select the left merge candidate, and partition P_1 tends to select the above merge candidate.

Fig. 6 and Fig. 7 show the number of candidate merges for GPM modes 12 and 18. These modes, 12 and 18, are characterized by distinct angles: mode 12 has a 45-degree angle, while mode 18 is horizontal. In both cases, P_0 has a high selection rate for the above merge candidates, whereas P_1 has a high selection rate for the left merge candidates. The reason for the different selection rates is thought to be the presence of partitions with the same movement on each of the long adjacent sides. Despite the variances in angular and position between GPM modes 12 and 18, both have the same trend in the selection rate of merge candidates for P_0 and P_1. In summary, we can conclude that the merge candidates are more likely to have longer contact of lengths with adjacent blocks. Other merge candidates, such as those at the bottom-left, above-right, and above-left, are infrequently selected, and no clear trend can be confirmed.

Fig. 8 and Fig. 9 show the numbers of merge candidates in GPM mode 27 and GPM mode 55 within Class A, respectively. Both modes, 27 and 55, represent diagonal partitioning of the same angle but at different positions. Both partitions P_0 and P_1 in GPM mode 27 are adjacent to the above and left blocks, and the selection of the above and left merge candidates remains relatively balanced across both partitions. In GPM mode 55, P_0 is not adjacent to the above or left block, resulting in a less biased selection distribution of merge candidates compared to modes 0, 12, and 18. In contrast, P_1 is adjacent to all of them, leading to a more dispersed selection rate than P_0. The position of the GPM partition appears to have minimal influence on the merge candidate selection rate.

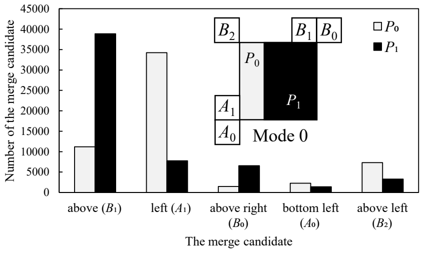


Fig. 5 Number of merge candidates in GPM mode 0 when Classes A1 and A2 are encoded by ECM 4.0.

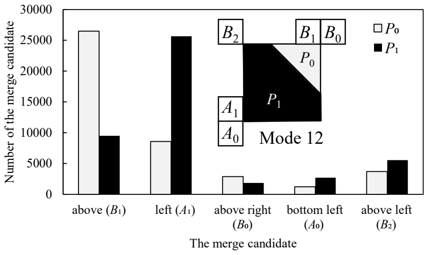


Fig. 6 Number of merge candidates in GPM mode 12 when Classes A1 and A2 are encoded by the ECM 4.0.

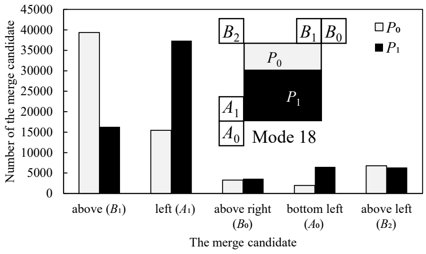


Fig. 7 Number of merge candidates in GPM mode 18 when Classes A1 and A2 are encoded by the ECM 4.0.

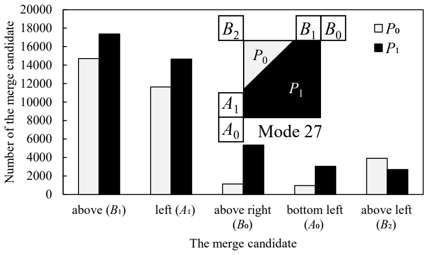


Fig. 8 Number of merge candidates in GPM mode 27 when Classes A1 and A2 are encoded by the ECM 4.0.

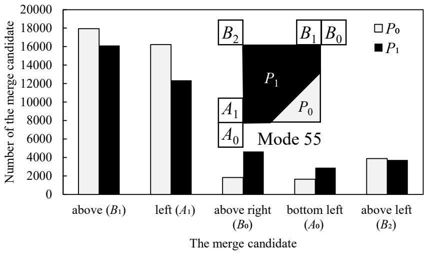


Fig. 9 Number of merge candidates in GPM mode 55 when Classes A1 and A2 are encoded by the ECM 4.0.

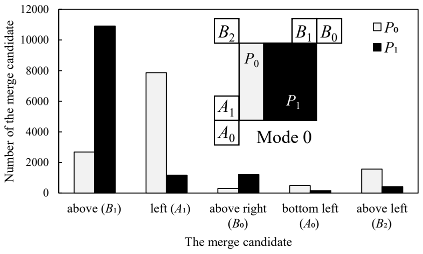


Fig. 10 Number of merge candidates in GPM mode 0 when Classes C and D are encoded by ECM 4.0.

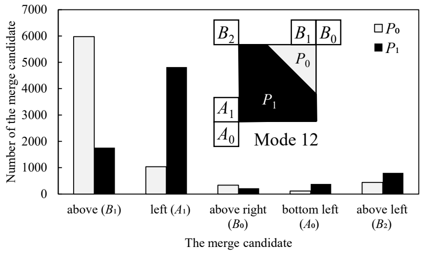


Fig. 11 Number of merge candidates in GPM mode 12 when Classes C and D are encoded by ECM 4.0.

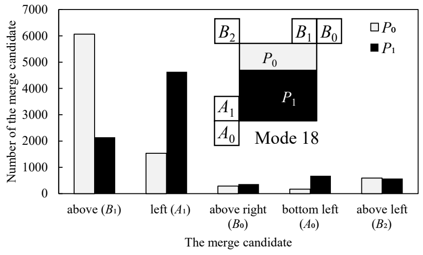


Fig. 12 Number of merge candidates in GPM mode 18 when Classes C and D are encoded by ECM 4.0.

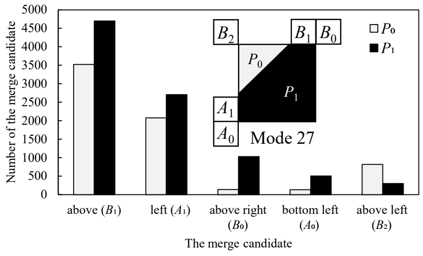


Fig. 13 Number of merge candidates in GPM mode 27 when Classes C and D are encoded by the ECM 4.0.

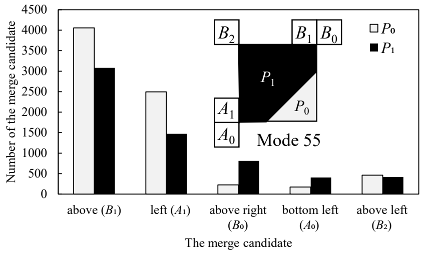


Fig. 14 Number of merge candidates in GPM mode 55 when Classes C and D are encoded by the ECM 4.0.

To investigate the influence of the resolution on the merge candidates, Fig. 10 through Fig. 14 show the numbers of merge candidates in GPM modes 0, 12, 18, 27, and 55 for Classes C and D, respectively. Class C has a resolution of 832 × 480, while Class D has a resolution of 416 × 240. The overall trend is the same as that for Class A. Comparing the ratios of the above and left merge candidates in each mode, it is evident that the ratio of Classes C and D exceeds that of Class A for all the modes. For example, comparing A_1 and B_1, which have high selection rates, the ratio of the above and the left merge candidates of P_1 in Fig. 10 is 9.4 (=10,912/1,166) for Classes C and D, while the same ratio is 5.0 (=38,867/7,773) for Class A in Fig. 5. Since the ratio is 87% larger for Classes C and D, we can say that Classes C and D merge candidates are more strongly influenced by the partitioning shape of the GPM. The effects observed in Classes C and D are explained by their low resolution and the reduced spatial gap between adjacent blocks in the foreground or background.

4. Proposed method

4.1 Merge candidate list construction based on the partition

We propose constructing the merge candidate list using only the motion vectors close to each GPM partition. Fig. 4 shows the five adjacent blocks of the coding block. Specifically, it is constructed by adding merge candidates to the list in the following order, following to the conventional merge list construction order:

1. Add the motion vectors of B_1 to the merge candidate list for the partition adjacent to the above block boundary of the current CU. This is determined by checking whether the region is connected to any samples in the above block.

2. Add the motion vectors of A_1 to the merge candidate list for the partition adjacent to the left block boundary of the current CU. This is determined by checking whether the region is connected to any samples in the left block.

3. Add the motion vectors of B_0 to the merge candidate list for the partition adjacent to the above-right block boundary of the current CU. This is determined by checking whether the region is connected to samples in the above-right block (B_0).

4. Add the motion vectors of A_0 to the merge candidate list for the partition adjacent to the bottom-left block boundary of the current CU. This is determined by checking whether the region is connected to samples in the bottom-left block (A_0).

5. Add the motion vectors of B_2 to the merge candidate list for the partition adjacent to the above-left block boundary of the current CU. This is determined by checking whether the region is connected to samples

Table 2 Example of proposed GPM merge candidate for each partition.

Marge candidate	mode 0		mode 12		mode 18		mode 27		mode 55	
index	P_0	P_1	P_0	P_1	P_0	P_1	P_0	P_1	P_0	P_1
0	B_1	B_1	B_1	B_1	B_1	A_1	B_1	B_1	B_1	B_1
1	A_1	B_0	B_0	A_1	A_1	A_0	A_1	A_1	A_1	A_1
2	A_0	-	-	A_0	B_0	-	B_2	B_0	B_0	B_0
3	B_2	-	-	B_2	B_2	-	-	A_0	A_0	A_0
4	-	-	-	-	-	-	-	-	$B2$	B_2

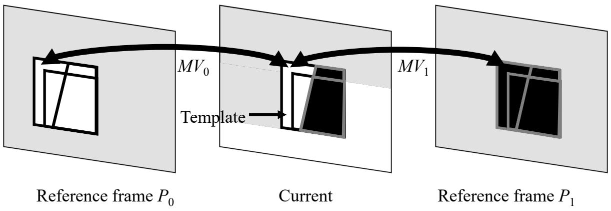


Fig. 15 Proposed reference frame selection by template matching.

Table 3 CTC test sequences from Class A to F categorized by resolution, frame rates, and bit depth.

Class	Resolution [sample]	Frame rate [FPS]	Bit depth [bit]
A1	3840 × 2160	60 or 30	10
A2	3840 × 2160	60 or 50	10
B	1920 × 1080	60 or 50	10 or 8
C	832 × 480	60, 50 or 30	8
D	416 × 240	60, 50 or 30	8
E	1280 × 720	60	8
F	832 × 480 - 1920 × 1080	20, 30, 50 or 60	8

in the above-left block (B_2).

For example, partition P_1 in mode 12 is determined to be adjacent to the above, left, above-left and bottom-left and not adjacent to the above-right. In mode 0, partition P_1 has a merge candidate list consisting of B_1 and B_0, and partition P_0 in mode 0 has a merge candidate list consisting of B_1, A_1, A_0, and B_2, as shown in Table 2. Note that the merge candidate list is not constrained in the case of partitions that are not directly adjacent to the left or above blocks.

4.2 Reference frame selection by template matching

We propose an adaptive method to select the better merge candidate from the two motion vectors without explicit signaling. The proposed method employs template matching to determine the list L_0 or L_1, similar to the conventional method [17]. The main difference is that template matching is applied to both L_0 and L_1 in the merge candidate list to select better motion vectors. Fig. 15 shows the template and comparables for each partition. For instance, the template for partition P_0 is partially above and completely left of the adjacent L shape. The template used four lines of samples adjacent to the partition. The list L_0 or L_1 is selected as follows.

1. Adjacent samples of each partition are used as templates.
2. For both L_0 and L_1 in the merge candidate list, the sum of the absolute difference (SAD) between the sample of the template and the sample at the referenced collocation is calculated.
3. The candidate with the minimal SAD is selected as the merge candidate without signaling.

5. Experimentation

5.1 Experimental conditions

The performance of the proposed method was compared using ECM 4.0 [15, 19] as a baseline. Although the ECM is updated quarterly, the universal value of the proposed method (academic novelty and progressiveness) is not lost due to the extension methods of GPM adopted by ECM between version 4.0 and the latest version 12.0 at the time of submission. The following GPM extension techniques were adopted in ECM between versions 4.0 and 12.0:

Regression-based GPM [20] This technique derives GPM coefficients through regression from reference samples.

GPM Affine [21] This technique applies affine motion compensation in GPM.

Bi-directive GPM [22] This technique applies bidirectional motion compensation for each region in GPM.

To compare the coding efficiency of multiple methods, JVET defined the common test conditions (CTC) [23] for several configurations. The CTC was defined as random access (RA) for entertainment applications with a 1-second random access cycle and low delay B (LDB) for interactive applications such as teleconferencing.

The CTC test sequences are classified into classes A through F for distinct resolutions and content types. Classes A to E consisted of camera-captured content, whereas Class F included screen content. Table 3 shows the resolutions, frame rates and bit depths of the test sequences. Certain test sequence classes are not evaluated with specific configurations to align the test sequence with the application scenarios. The low-resolution Class D and screen content Class F were excluded from the overall average calculations because they were deemed not to be relevant to the primary objective. Furthermore, CTC does not evaluate Class E for teleconferences under the RA configuration because of the delay due to temporally backward prediction. Moreover, CTC does not evaluate high-resolution Class A under the

Table 4 Performance of the proposed method under RA [%].

Class	Y	U	V	EncT	DecT
A1	-0.12	-0.26	-0.08	102.3	100.8
A2	-0.13	-0.18	-0.21	103.0	100.9
B	-0.12	-0.09	-0.09	104.4	100.4
C	-0.16	-0.21	-0.22	106.0	101.1
E	-	-	-	-	-
Overall	-0.13	-0.18	-0.15	104.1	100.8
D	-0.11	-0.15	-0.02	105.8	101.1
F	-0.11	-0.23	-0.18	104.2	100.3

Table 5 Performance of the proposed method under LDB [%].

Class	Y	U	V	EncT	DecT
A1	-	-	-	-	-
A2	-	-	-	-	-
B	-0.07	-0.28	-0.19	103.2	100.4
C	-0.19	-0.28	-0.43	106.1	102.2
E	-0.26	-0.86	-0.21	105.3	99.9
Overall	-0.16	-0.42	-0.27	104.7	100.8
D	-0.15	-0.55	-0.26	105.7	102.6
F	-0.12	-0.45	-0.17	104.8	101.5

LDB configuration due to the high resolution of Class A for teleconferencing. Four quantization parameter (QP) values of 22, 27, 32, and 37 were used to generate the different rate points. The coding efficiency was evaluated using BD-rate [24, 25]. The BD-rate is well known as an evaluation metric for quantifying the difference in bitrate for equivalent levels of the peak signal-to-noise ratio (PSNR), and a negative BD-rate indicates bitrate savings and better coding efficiency. The BD-rate is evaluated for each Y, U, and V, each representing luminance and two representing chrominance. The GPM boundary width τ is set to 4. The complexity was evaluated by the ratio of the encoding time (EncT) and the decoding time (DecT) to the baseline.

5.2 Experimental results

Tables 4 and 5 show the BD-rates of the proposed method under the RA and LDB configurations, respectively. The overall BD-rates of the proposed method are -0.13% for RA and -0.16% for LDB. This improvement indicates that the proposed method can place the better merge candidate at the top of the merge candidate list. The reason why the LDB gain is greater than the RA gain is because GPM is more likely to be selected for in LDB than in RA. RA can refer to both past and future frames. In contrast, LDB is limited to referencing only past frames, leading to relatively low accuracy in LDB motion compensation, which is partitioned and compensated by GPM. A previous study reported that GPM is more frequently selected in LDB than in RA [6], and the proposed method was shown to improve the coding efficiency of GPM with the same tendency.

A comparison of the results among classes reveals a remarkable improvement of -0.16% in Class C for RA and -0.26% in Class E for LDB. We were able to confirm that the influence of the GPM merge candidate list is greater in classes with a relatively low resolution, as analyzed in Section 3. However, it is important to note that Classes A1 and A2, which represent high-resolution sequences, also show improvements. This indicates that while the proposed method is beneficial across all resolutions, the degree of improvement can vary depending on the specific characteristics of each sequence.

Table 6 GPM Selection Rate for RA and LDB configurations [%].

QP	RA		LDB	
	ECM4	Proposal	ECM4	Proposal
22	9.1	9.9	11.3	12.1
27	11.8	12.7	15.8	16.8
32	12.9	13.9	17.5	18.4
37	13.6	14.7	18.1	19.1

Table 7 Sequence-level performance of the proposed method under RA and LDB configurations [%].

Class	Sequence	RA	LDB
A1	Tango2	-0.22	-
	FoodMarket4	-0.08	-
	Campfire	-0.06	-
A2	Catrobot1	-0.18	-
	DaylightRoad2	-0.13	-
	ParkRunning3	-0.08	-
B	MarketPlace	-0.12	-0.01
	RitualDance	-0.10	-0.10
	Cactus	-0.13	-0.12
	BasketballDrive	-0.18	-0.12
	BQTerrace	-0.10	0.02
C	BasketballDrill	-0.19	-0.29
	BQMall	-0.19	-0.27
	PartyScene	-0.12	0.00
	RaceHorsesC	-0.13	-0.18
D	BasketballPass	-0.06	-0.13
	BQSquare	-0.02	-0.15
	BlowingBubbles	-0.19	-0.21
	RaceHorses	-0.19	-0.11
E	FourPeople	-	-0.19
	Johnny	-	-0.42
	KristenAndSara	-	-0.18
F	BasketballDrillText	-0.19	-0.05
	ArenaOfValor	-0.19	-0.16
	SlideEditing	-0.04	-0.05
	SlideShow	-0.05	-0.22

To understand the specific factors contributing to the improvement in coding efficiency, we analyzed the GPM selection rate. The selection rate was defined as the percentage of GPM blocks in all CUs in the inter slice. As shown in Table 6, the GPM selection rate was higher in the proposed method compared to the baseline. The GPM selection rate in Table 6 reflects the results of rate-distortion optimization. The proposed method consistently shows higher selection rates across various QP levels for both RA and LDB configurations compared to the baseline. This increase in GPM selection rate in the proposed method is due to its ability to reduce the prediction error of GPM without increasing the signaling overhead, thereby reducing the bitrate. By analysising of these factors, we can attribute the overall improvement in coding efficiency to the increased GPM selection rate and the reduction in prediction error.

Table 7 shows the BD-rate of each sequence.

The proposed method significantly improves the Tango2 sequence by -0.22% under RA and the Johnny sequence by -0.42% under LDB. Sequences such as Tango2, BQMall, BasketballDrill, and Johnny, which have relatively large gains in the original GPM, show improved coding efficiency due to the increased effectiveness of the proposed method. This improvement is attributed to the high correlation between the motion vectors. In contrast, the BD-rate for BQTerrace under LDB decreased due to a low correlation between adjacent motion vectors. From these observations, a higher correlation results in better coding efficiency for the proposed method. For SlideEditing, which features distinct edges, GPM underperforms due to blurring caused by blending [6]. Nonetheless, modest improvements were observed in both the RA and LDB configurations by constructing suitable merge candidate lists with the proposed method.

To address the high variability among sequences and the observed differences between RA and LDB, error bars were added to the performance graphs to indicate the variability and significance of the results. Figs. 16 and 17 show the average BD-rates and standard deviations for each class under RA and LDB configurations.

The error bars were calculated as follows: for each class, the mean BD-rate was computed, and the standard deviation was calculated to represent the variability within each class. The variability in the error bars is due to the inherent characteristics of GPM, which is often used in sequences that contain object boundaries with motion. Therefore, the performance improvement of GPM is highly sequence dependent [6]. The proposed method also shows performance improvements in sequences where GPM is effective, following a similar trend to the original GPM. This suggests that the effectiveness of the proposed method depends on the characteristics of each sequence, which leads to higher variability, especially in the LDB configuration. Consequently, the effectiveness of the proposed method varies significantly across different sequences, indicating that the benefits of the method are more significant in sequences with certain characteristics, particularly in the presence of moving object boundaries.

The differences between RA and LDB configurations are evident in the error bar analysis. For Class D, the error bars are large for RA but small for LDB, indicating higher variability in RA. This suggests that the performance of the proposed method on Class D sequences is more stable under the LDB configuration. Conversely, Class B shows similar error bars for both RA and LDB, indicating consistent performance across configurations. Class C, on the other hand, shows the opposite trend to Class D, with smaller error bars for RA and larger ones for LDB, highlighting more stable performance in RA. These differences indicate the importance of considering the specific characteristics of each sequence and configuration when evaluating the effectiveness of the proposed method.

Fig. 18 shows the PSNR curves for the proposed method and ECM 4.0 in Tango2 with the highest coding efficiency under the RA configuration. The proposed method reduces the bitrate with the same image quality as ECM 4.0 due to the reduction in residuals by the improved prediction accuracy of the GPM. The other test sequences showed the same tendency.

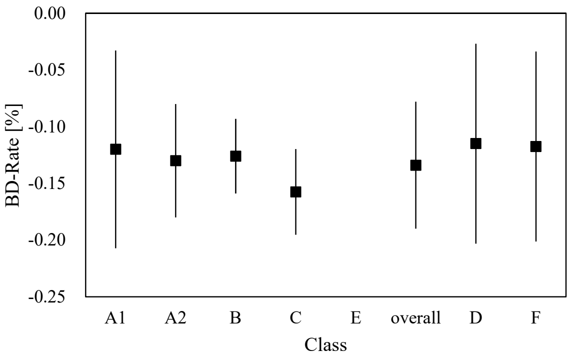


Fig. 16 BD-rates with error bars across classes under the RA configuration.

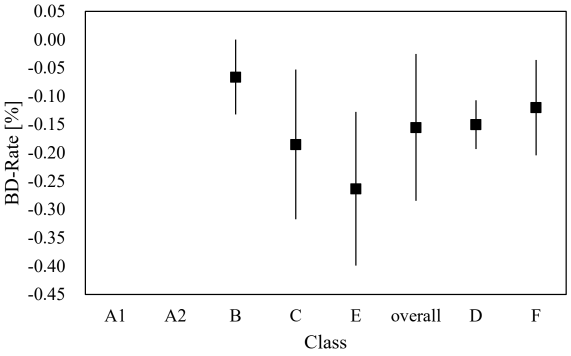


Fig. 17 BD-rates with error bars across classes under the LDB configuration.

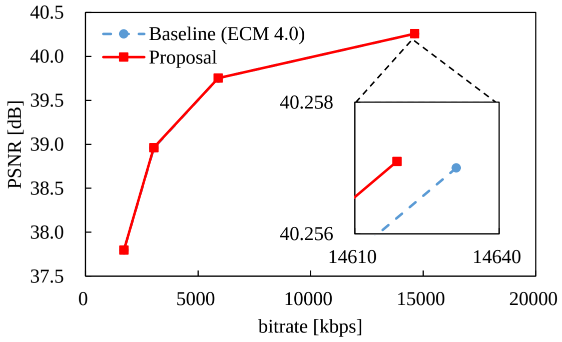


Fig. 18 PSNR curves for the proposed method and the baseline method (ECM 4.0) in the Tango2 test sequence under the RA configuration.

Regarding computational complexity, the proposed method increases the overall EncT to 104.7% but suppresses the overall DecT to 100.8%. The baseline can compute the cost of rate-distortion optimizations (RDO) [26] for two

Table 8 The overall BD-rate (Y) of ablation study under RA configuration [%]. The best result is shown in bold text.

	Subsection 4.1		Subsection 4.2	BD-rate (Y)
No.	above and left	above-left		
1	x			-0.07
2	x	x		-0.08
3	x		x	-0.08
4	x	x	x	**-0.13**

partitions with a single motion compensation because the merge candidates are identical. Conversely, the proposed method increases the number of cost calculations because the merge candidates are different for each partition. The increase in RDOs increases EncT. Template matching is required for L_0 or L_1 selection between the encoding and decoding processes, but a small DecT indicates an insignificant computational load. The conventional method [8] increases DecT to 106% due to its dependency on template matching for motion estimation. However, the proposed method has a small impact on decoding because only the merge candidates are targeted for template matching. Compared to the coding efficiency of the conventional methods described in Subsection 2.1, the tradeoff between the gain and processing time with the proposed method is reasonable. In addition, the proposed method differs from conventional methods, so the gains can be expected to be accumulated by the combination.

5.3 Ablation study

Table 8 shows the coding efficiency for different combinations of construction processes for merge candidates.

Condition No. 1: This is an initial condition that doesn't select the upper left block (B_2) or the reference frame. It only considers the remaining blocks (B_1, B_0, A_1, A_0) to construct merge candidates. It achieves a BD-rate of -0.07%.

Condition No. 2: This condition evaluates the impact of including the above-left block (B_2) in the merge candidate list. As shown in Figs. 13 and 14, the selection rate of B_2 is relatively low even in the region adjacent to B_2. Therefore, whether B_2 is necessary should be evaluated. The comparison shows that enabling the above-left block (No. 2) results in a slight BD-rate improvement (-0.08% compared to -0.07%). Although the improvement is minimal, it suggests that including the above-left block B_2 is not harmful and may provide a small benefit to coding efficiency by providing additional spatial information.

Condition No. 3: This condition evaluates the effect of reference frame selection by template matching. Comparing No. 1 and No. 3, the reference frame selection (No. 3) also results in a slight BD-rate improvement (-0.08% compared to -0.07%). This indicates that reference frame selection has a marginal contribution to improving coding efficiency by enhancing prediction accuracy. Although the improvement is small, it suggests that incorporating reference frame selection is not disadvantageous and can offer incremental benefits to overall performance.

Condition No. 4: This condition evaluates the combined effects of all methods. Condition No. 4 achieves the best BD-rate improvement (-0.13%). This indicates that using motion vectors close to each GPM partition, along with an adaptive method to select the better merge candidate from the two motion vectors, leads to the most effective enhancement in coding efficiency. The combined application of all methods results in a more efficient coding process than applying each method independently.

6. Conclusion

We proposed an adaptive method for GPM to construct merge candidate lists to improve the coding efficiency over VVC. Our analysis shows the selection rate of merge candidates for various GPM modes, revealing a dependency on the adjacent block lengths for each partition. Additionally, this paper presented a method to identify GPM reference frames without signaling. The experimental results showed that the proposed method provides additional bitrate savings of 0.13% and 0.16% for the RA and LDB configurations, respectively, compared to those of the ECM 4.0 reference software. In future work, we will resolve conflicts with the contribution [17] restricting the GPM by merge candidates.

Acknowledgments

These research results were obtained from the commissioned research (No. 06801) by National Institute of Information and Communications Technology (NICT), Japan.

References

[1] "Recommendation H.266: Versatile video coding." `https://www.itu.int/rec/T-REC-H.266`, 2020. [visited on Feb. 22, 2024].

[2] "Recommendation H.265: High efficiency video coding." `https://www.itu.int/rec/T-REC-H.265`, 2020. [visited on Feb. 22, 2024].

[3] B. Bross, Y. 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. DOI: 10.1109/TCSVT.2021.3101953.

[4] W. Hamidouche, T. Biatek, M. Abdoli, E. François, F. Pescador, M. Radosavljević, D. Menard, and M. Raulet, "Versatile video coding standard: A review from coding tools to consumers deployment," IEEE Consumer Electronics Magazine, vol.11, no.5, pp.10–24, 2022. DOI: 10.1109/MCE.2022.3144545.

[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. DOI: 10.1109/TCSVT.2021.3101212.

[6] 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, 2021. DOI: 10.1109/TCSVT.2020.3040291.

[7] X. Meng, X. Zhang, C. Jia, X. Li, S. Wang, and S. Ma, "Edge-directed geometric partitioning for versatile video coding," 2020 IEEE

International Conference on Multimedia and Expo (ICME), pp.1–6, 2020. DOI: 10.1109/ICME46284.2020.9102781.

[8] 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, 2022. DOI: 10.1109/TIP.2021.3126420.

[9] Y. Kidani, H. Kato, K. Kawamura, and H. Watanabe, "Geometric partitioning mode with inter and intra prediction for beyond versatile video coding," IEICE Transactions on Information and Systems, vol.E105-D, no.10, pp.1691–1703, 2022. DOI: 10.1587/transinf.2022PCP0005.

[10] M. Lee, S.J. Oh, and D. Sim, "Decoder side GPM weight matrix derivation method for SCC in VVC," 2022 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), pp.1–4, 2022. DOI: 10.1109/ICCE-Asia57006.2022.9954773.

[11] H. Kato, Y. Kidani, K. Kawamura, and S. Naito, "Adaptive boundary width of geometric partitioning mode for beyond versatile video coding," 2022 IEEE International Conference on Visual Communications and Image Processing (VCIP), pp.1–5, 2022. DOI: 10.1109/VCIP56404.2022.10008865.

[12] Y. Kidani, H. Kato, K. Unno, and K. Kawamura, "Adaptive width for GPM blending area," Tech. Rep. JVET-Z0059, JVET, 2022.

[13] Y. Kidani, H. Kato, K. Unno, K. Kawamura, N. Yan, X. Xiu, W. Chen, H.J. Jhu, C.W. Kuo, and X. Wang, "GPM adaptive blending," Tech. Rep. JVET-AA0058, JVET, 2022.

[14] Z. Deng, K. Zhang, and L. Zhang, "Geometry partitioning with motion vector difference for video coding," 2022 IEEE International Conference on Image Processing (ICIP), pp.2271–2275, 2022. DOI: 10.1109/ICIP46576.2022.9898040.

[15] "Algorithm description of enhanced compression model 4 (ECM-4.0)." `https://vcgit.hhi.fraunhofer.de/ecm/ECM/-/tree/ECM-4.0`, 2022. [visited on Feb. 22, 2024].

[16] V. Seregin, J. Chen, F.L. Léannec, and K. Zhang, "JVET AHG report: ECM software development (AHG6)," Tech. Rep. JVET-Z0006, JVET, 2022.

[17] C.C. Chen, H. Huang, Y. Zhang, Z. Zhang, Y.J. Chang, V. Seregin, and M. Karczewicz, "Template matching based reordering for GPM split modes," Tech. Rep. JVET-Y0135, JVET, 2022.

[18] X. Wang, Y.W. Chen, X. Xiu, and T.C. Ma, "An improved method for triangle merge list construction," Tech. Rep. JVET-N0340, JVET, 2019.

[19] M. Coban, F.L. Léannec, K. Naser, and J. Ström, "Algorithm description of enhanced compression model 4 (ECM 4)," Tech. Rep. JVET-Y2025, JVET, 2022.

[20] P. Bordes, K. Reuzé, F. Galpin, F. Urban, K. Naser, F.L. Leannec, and E. Francois, "EE2-2.11: Regression-based GPM blending (tests a,b,c)," Tech. Rep. JVET-AG0112, JVET, 2024.

[21] K. Zhang, Z. Deng, and L. Zhang, "EE2-2.10: GPM with affine prediction," Tech. Rep. JVET-AG0164, JVET, 2024.

[22] R. Yu, P. Wennersten, J. Enhorn, and K. Andersson, "Ee2-3.2: Bi-predictive gpm," Tech. Rep. JVET-AE0046, JVET, 2023.

[23] M. Karczewicz and Y. Ye, "Common test conditions and evaluation procedures for enhanced compression tool testing," Tech. Rep. JVET-Y2017, JVET, 2022.

[24] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," Tech. Rep. VCEG-M33, VCEG, 2001.

[25] K. Andersson, F. Bossen, J.R. Ohm, A. Segall, R. Sjöberg, J. Ström, and G.J. Sullivan, "Summary information on BD-rate experiment evaluation practices," Tech. Rep. JVET-Q2016, JVET, 2020.

[26] G.J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," IEEE signal processing magazine, vol.15, no.6, pp.74–90, 1998. DOI: 10.1109/79.733497.

Haruhisa Kato received B.E. and M.E. degrees in electronical and electronic engineering from Kobe University in 1997 and 1999, respectively. He received a Ph.D. from Graduate School of Engineering, Tokyo Institute of Technology in 2009. In 1999, he joined KDD Co. Ltd. His research interests include image processing and augmented reality. He is a Senior Expert of the Advanced Technology Laboratory at KDDI Research, Inc.

Yoshitaka Kidani received B.E. and M.E. degrees in Engineering from Nagoya University, in 2010 and 2012. He joined KDDI Corporation in 2012. He is currently a research engineer at KDDI Research, Inc. He has been involved with the development of VVC standards under JVET. His research interests include video coding and multimedia distribution.

Kei Kawamura received his B.E., M.Sc., and Ph.D. degrees in Global Information and Telecommunication Studies from Waseda University, Japan, in 2004, 2005, and 2013, respectively. He joined KDDI in 2010. He has been involved with the development of HEVC and VVC standards under JCT-VC and JVET. He is currently engaged in the research and development of a video coding system at KDDI Research, Inc. His research interests include image and video processing, video coding, and multimedia distribution.