electronics

MDPI

Article

Adjacent Intra Mode-Based Prediction Blending for VVC and Beyond-VVC

Kyuri Kim [1,†], Gihwa Moon [1,2,†] and Jae-Gon Kim [1,2,*]

[1] Department of Electronics Engineering, Korea Aerospace University, Goyang 10540, Republic of Korea; kyrkim@kau.kr (K.K.); ghmoon@kau.kr (G.M.)
[2] Department of Smart Air Mobility, Korea Aerospace University, Goyang 10540, Republic of Korea
* Correspondence: jgkim@kau.ac.kr
† These authors contributed equally to this work.

Abstract

In Versatile Video Coding (VVC), conventional angular intra prediction uses a single prediction mode, which may limit prediction accuracy for blocks with spatially varying directional structures. This paper proposes adjacent intra mode-based prediction blending (AIM-PB), which directly exploits neighboring intra prediction modes (IPMs) available at five predefined adjacent positions. AIM-PB combines the corresponding angular predictions with Planar prediction using adaptive weights derived from neighboring IPM information. The predictor modes and blending weights are obtained without requiring gradient analysis or template-cost evaluation. Experiments using the VVC Test Model (VTM) 23.5 show average luma Bjøntegaard delta rate (BD-rate) reductions of 0.31% and 0.15% under the All-Intra (AI) and Random-Access (RA) configurations, respectively, with a maximum reduction of 0.60% for *FoodMarket4* under AI. Further evaluation using NextSoftware2 (NX2) shows additional luma coding gains in the tested beyond-VVC environment. These results show that AIM-PB provides luma coding gains, with average decoding-time ratios of approximately 100% relative to the corresponding anchors on the tested platforms.

Keywords: Versatile Video Coding (VVC); intra prediction; prediction blending; neighboring intra modes; beyond-VVC

1. Introduction

Versatile Video Coding (VVC), standardized as ITU-T H.266 and ISO/IEC 23090-3, provides substantially improved compression efficiency over previous video coding standards by introducing advanced block partitioning, prediction, transform, and entropy coding techniques [1,2]. Following the completion of VVC, the Joint Video Experts Team (JVET) has continued investigating video coding technologies with capability beyond VVC. In July 2026, JVET issued the Joint Call for Proposals on video compression with capability beyond VVC, marking the transition toward the development of a next-generation video coding standard [3].

Intra prediction reduces spatial redundancy by predicting the current block from reconstructed neighboring samples [4]. Conventional angular intra prediction generates a prediction signal using a single angular mode associated with a specific prediction direction. However, a single angular predictor may be insufficient for blocks exhibiting multiple or locally varying directional characteristics. Accordingly, recent intra prediction approaches have investigated the combination of multiple prediction signals derived from different pre-

Check for updates

Academic Editor: George A. Tsihrintzis

Received: 5 October 2026
Revised: 1 November 2026
Accepted: 17 November 2026
Published: 27 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

diction modes to exploit complementary prediction characteristics and improve prediction performance [5–8].

Some existing multi-predictor intra prediction approaches determine prediction modes or blending weights by analyzing reconstructed neighboring samples or evaluating prediction errors over reconstructed templates. However, available neighboring intra-coded blocks already provide intra prediction modes (IPMs) that can convey local directional information. These IPMs are available at both the encoder and decoder and are already exploited by occurrence-based intra coding (OBIC) [8]. This motivates examining whether additional coding gains can be obtained using a limited set of adjacent IPMs without block-area-based occurrence accumulation or candidate ranking based on a histogram of occurrences (HoC).

Based on this observation, this paper proposes adjacent intra mode-based prediction blending (AIM-PB). AIM-PB directly reuses angular IPMs from five predefined adjacent positions around the current coding unit (CU) and combines the corresponding angular predictions with Planar prediction using adaptive integer weights. The predictor modes and blending weights are derived directly from neighboring IPM information, without additional gradient analysis or template-cost evaluation.

AIM-PB is implemented in the VVC Test Model (VTM) 23.5 and evaluated under the JVET common test conditions (CTC). Additional experiments using NextSoftware2 (NX2) evaluate its performance in a coding environment incorporating advanced intra prediction tools.

The main contributions of this work are as follows:

- We propose AIM-PB, an intra prediction blending method that collects at most one angular IPM from each of five predefined adjacent positions and uses all unique collected angular modes together with Planar prediction. Angular weights are derived from position-wise occurrence counts, while the Planar weight is determined by the number of positions providing valid angular IPMs. This design avoids block-area-based HoC construction and HoC-based candidate ranking.
- We evaluate the coding efficiency and runtime of AIM-PB and analyze its selection behavior and predictor composition. We also compare design variants with different angular weighting schemes, Planar prediction settings, and neighboring-position configurations.
- We investigate various AIM-PB integration configurations in NX2 and evaluate the coding performance and runtime of a representative configuration under the JVET CTC.

The remainder of this paper is organized as follows. Section 2 reviews VVC intra prediction and related beyond-VVC intra prediction techniques. Section 3 describes the proposed AIM-PB method. Section 4 presents the experimental setup, VTM coding performance, coding-behavior and design analyses, and additional NX2 evaluations. Section 5 discusses the main findings and the trade-off between coding efficiency and runtime. Section 6 concludes this paper.

2. Background and Related Work

2.1. Intra Prediction in VVC

VVC supports 67 luma intra prediction modes, consisting of Planar, DC, and 65 angular modes [4]. Angular intra prediction generates prediction samples from reconstructed neighboring reference samples according to the direction specified by the selected prediction mode. Planar prediction generates a smooth two-dimensional prediction from neighboring reference samples.

VVC additionally incorporates several tools for improving intra prediction efficiency. Multiple Reference Line (MRL) allows angular prediction to use reference lines other than the adjacent line [9,10]. Intra subpartition (ISP) mode divides a coding block into smaller subpartitions for sequential intra prediction and residual coding [11,12]. Matrix-based in-

tra prediction (MIP) generates a prediction block from downsampled boundary reference samples using matrix-vector multiplication, followed by upsampling when required [4,13,14].

2.2. Intra Prediction Tools Explored Beyond VVC

Decoder-side mode derivation and prediction fusion have been investigated to further improve intra prediction beyond VVC. These methods differ in the neighboring information used to derive prediction modes and blending weights.

Decoder-side intra mode derivation (DIMD) applies horizontal and vertical Sobel filters to reconstructed neighboring samples. The gradient directions are mapped to angular IPMs, and the corresponding gradient magnitudes are accumulated in a histogram of gradients (HoG). The HoG is then used to select angular prediction modes and derive their blending weights [5,15].

Template-based intra mode derivation (TIMD) uses a template formed by reconstructed samples on the top and left sides of the current block. Each tested IPM predicts this template using its reconstructed reference samples. The predicted template is compared with the reconstructed template to obtain a matching cost. These costs guide mode selection and fusion-weight derivation for the current block [6,16].

Related studies investigate alternative prediction-fusion schemes [7], fusion involving intra template matching prediction [17,18], and the incorporation of additional predictor types into TIMD fusion [19,20]. TIMD-Merge reuses prediction information from neighboring TIMD-coded blocks, while retaining template-cost-based candidate selection [21]. Luo et al. proposed separate fusion-based methods for adaptive template matching and linear prediction in VVC [22].

OBIC constructs the HoC from neighboring IPMs using block-area-based contributions. Multiple IPMs associated with a neighboring prediction may be included in the HoC. Up to five angular modes are selected based on their HoC values, and the corresponding prediction blocks are blended with Planar prediction or a block-vector-based predictor [8]. The angular blending weights are derived from the HoC values [8]. Weighted OBIC further incorporates neighboring blending weights, a distance-dependent factor, and location-dependency information into the HoC derivation [23].

2.3. Differences Between AIM-PB and Related Methods

Table 1 compares the information used for mode derivation, the construction of the prediction-mode set, and the derivation of blending weights.

Unlike DIMD's Sobel-based gradient analysis and regular TIMD's candidate-wise template prediction and cost evaluation, AIM-PB derives its predictor modes and weights directly from neighboring IPM observations. This difference concerns mode and weight derivation; reconstructed reference samples are still used to generate the angular and Planar prediction blocks.

OBIC and AIM-PB both reuse neighboring IPM information, but differ mainly in the number of neighboring locations examined and the prediction information reused from each neighbor. In the Enhanced Compression Model (ECM) 20.0 implementation [24], OBIC examines up to 31 neighboring CU candidates, including 13 adjacent and 18 non-adjacent candidates. After availability and duplicate checks, up to 20 neighboring CUs are retained for HoC construction. For each retained CU, OBIC may reuse multiple IPMs and their associated blending weights used for coding that CU.

In contrast, AIM-PB examines only five predefined adjacent positions and uses at most one eligible angular IPM at each position. When the corresponding neighboring CU uses multiple prediction modes, only one representative angular IPM is used for AIM-PB mode derivation, and neighboring blending weights are not used. Thus, AIM-PB uses a fixed five-

position collection process and position-wise occurrence counting, without non-adjacent mode collection, block-area-based HoC construction, or HoC-based candidate ranking.

A quantitative decoder-side operation analysis of DIMD, normal TIMD, OBIC, and AIM-PB in the tested JVET-AN0267-based NX2 environment is presented in Section 4.5.3.

Table 1. Comparison of representative intra prediction methods utilizing neighboring information in terms of input information, prediction-mode set construction, and blending-weight derivation.

Method	Information Used	Mode-Set Construction	Blending-Weight Derivation
DIMD [5,15]	Reconstructed neighboring samples	Sobel filtering, HoG accumulation, and angular-mode selection	Angular weights derived from HoG amplitudes
TIMD [6,16]	Reconstructed top/left template and its reference samples	Template prediction with each tested IPM, followed by comparison with the reconstructed template and cost-based selection	Fusion weights derived from template costs
OBIC [8,23]	Neighboring IPMs and block areas from adjacent and non-adjacent blocks, including all IPMs specified for multi-IPM neighbors [a]; neighboring blending weights are additionally used in weighted OBIC [b]	Block-area-based HoC accumulation, followed by selection of up to five angular modes with the highest HoC values	Angular weights derived from HoC values using a DIMD-like weighting approach
AIM-PB	At most one available angular IPM from each of five adjacent positions; a representative IPM from an AIM-PB-coded neighbor	Merging of identical mode entries while retaining occurrence counts; all unique collected angular modes are used without further occurrence-based selection	Planar weight determined by the number of positions providing valid angular IPMs; angular weights proportional to position-wise occurrence counts

[a] The neighboring-mode sets specified in [8] contain up to five angular IPMs for DIMD and OBIC, up to two IPMs for TIMD, and two IPMs for spatial geometric partitioning mode (SGPM). All IPMs in the corresponding mode set are included in the HoC, and each included IPM contributes the area of the neighboring block to its HoC entry. [b] Weighted OBIC scales each area-based HoC contribution using the corresponding blending weight of the IPM in the neighboring predictor and a distance-dependent factor. The associated location-dependency state is also reused [23].

3. Proposed AIM-PB

Figure 1 illustrates the overall procedure of AIM-PB. The procedure for a luma CU is summarized as follows:

1. Collect eligible angular IPMs from five predefined adjacent positions in the order $T1$, $L1$, TL, $L2$, and $T2$.
2. Merge identical angular modes while retaining their occurrence counts and the number of valid positions N.
3. If $N = 0$, use Planar prediction and set Planar as the representative IPM. Otherwise, generate Planar and one angular predictor for each unique collected mode.
4. For $N > 0$, derive the blending weights and generate the blended prediction.
5. For $N > 0$, select the most frequently collected angular mode as the representative IPM, resolving ties by the predefined collection order.
6. Store the derived representative IPM when AIM-PB is selected for the current CU.

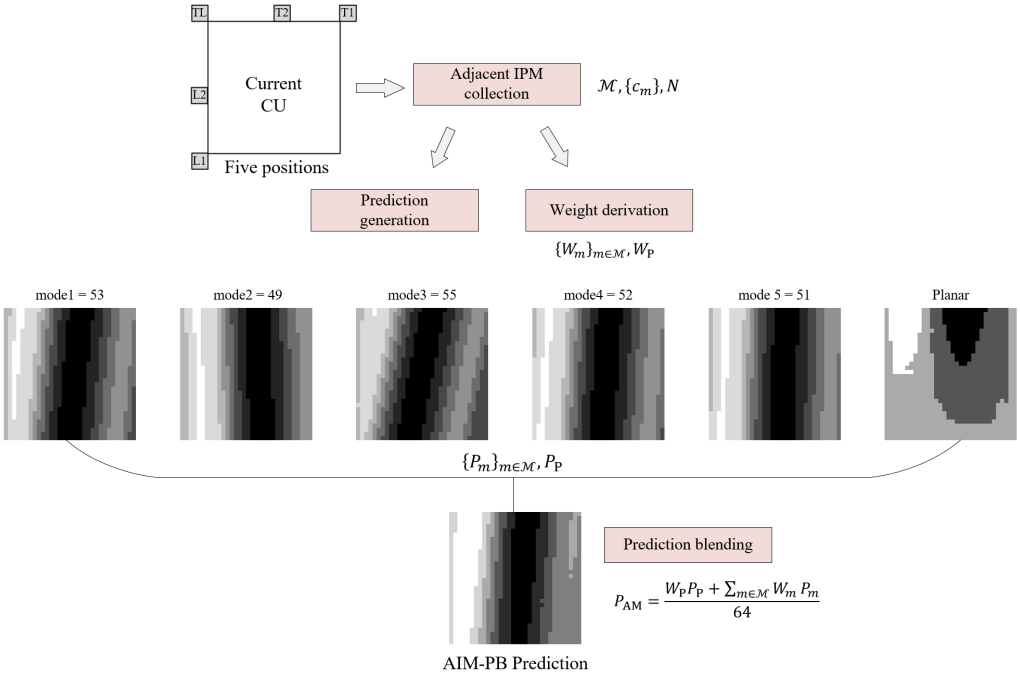


Figure 1. Overview of AIM-PB. Available angular IPMs are collected from five predefined adjacent positions, and identical mode entries are merged while retaining their occurrence counts. One prediction block is generated for each unique angular mode and blended with Planar prediction using occurrence-based weights.

3.1. Adjacent IPM Collection

For the current luma CU, five predefined adjacent positions are examined in the fixed order $T1$, $L1$, TL, $L2$, and $T2$, as illustrated in Figure 1. At most one available angular IPM is collected from the neighboring CU at each position. Planar and DC modes, as well as MIP-coded neighboring CUs, are excluded from mode collection.

Identical mode entries are merged to form a set of unique angular modes, denoted by $\mathcal{M}$, while retaining the occurrence count of each mode. Let N denote the number of positions providing valid angular IPMs, $K = |\mathcal{M}|$ the number of unique angular modes, and c_m the number of selected positions at which mode m is collected. These quantities satisfy

$$N = \sum_{m\in\mathcal{M}} c_m, \qquad 0 \le K \le N \le 5. \tag{1}$$

Occurrence counts in Equation (1) are accumulated by position rather than by distinct neighboring CU. Thus, multiple selected positions belonging to the same CU contribute separately, and a neighboring CU covering more sampled positions can contribute more to the occurrence count of its angular IPM. Only K angular prediction blocks are generated, regardless of repeated occurrences of the same mode.

For each AIM-PB-coded CU with $N > 0$, the angular mode with the largest occurrence count is selected and stored as the CU's representative IPM. Ties are resolved by selecting the mode encountered first in the predefined collection order, which prioritizes $T1$ and $L1$. For the same collected IPMs, the occurrence counts and blending weights are independent of this order. When the CU is subsequently referenced as a neighboring block, AIM-PB and other coding tools use the same stored representative IPM. The representative IPM is derived and stored identically at the encoder and decoder.

When no valid angular IPM is available, $N = K = 0$ and $\mathcal{M}$ is empty. In this case, the AIM-PB prediction defaults to Planar prediction, and the Planar mode is stored as the

CU's representative IPM. This provides a deterministic, non-directional fallback without requiring additional angular-mode derivation.

3.2. Prediction Generation and Weight Derivation

For each unique angular mode $m \in \mathcal{M}$, an angular prediction block P_m is generated using the conventional VVC angular prediction process with reconstructed neighboring reference samples [4]. A Planar prediction block P_{P} is also generated and included in the weighted blending.

AIM-PB uses nonnegative integer blending weights that sum to 64. The Planar weight W_{P} is determined by the number N of positions providing valid angular IPMs:

$$W_{\mathrm{P}}(N) = \begin{cases} 64, & N = 0, \\ 32, & N = 1, \\ 24, & N = 2, \\ 16, & N = 3, \\ 8, & N = 4, \\ 4, & N = 5. \end{cases} \tag{2}$$

The Planar weight schedule in Equation (2) is heuristically designed to decrease the Planar contribution as the number of positions providing valid angular IPMs (N) increases, thereby assigning greater total weight to angular predictions. A nonzero Planar contribution is retained for $1 \leq N \leq 5$. The selected values also ensure that $(64 - W_{\mathrm{P}}(N))/N$ is an integer, allowing the angular weights to be derived from the position-wise occurrence counts using integer scaling.

For $N \geq 1$, the weight assigned to angular mode m is

$$W_m = (64 - W_{\mathrm{P}}) \frac{c_m}{N}. \tag{3}$$

For a given CU, angular modes with larger occurrence counts receive larger blending weights according to Equation (3). The Planar and angular blending weights therefore satisfy Equation (4):

$$W_{\mathrm{P}} + \sum_{m \in \mathcal{M}} W_m = 64. \tag{4}$$

3.3. Prediction Blending

The final AIM-PB prediction is calculated as

$$P_{\mathrm{AM}} = \frac{W_{\mathrm{P}} P_{\mathrm{P}} + \sum_{m \in \mathcal{M}} W_m P_m}{64}, \tag{5}$$

where P_{P} denotes the Planar prediction and P_m denotes the angular prediction generated using mode m. For $N = 0$, the angular summation in Equation (5) is empty and $P_{\mathrm{AM}} = P_{\mathrm{P}}$. For $N \geq 1$, the prediction combines K unique angular predictors with Planar. The weighted blending operation is implemented using fixed-point integer arithmetic.

3.4. Encoder and Decoder Integration

At the encoder, AIM-PB is evaluated as an additional luma intra prediction candidate using full rate-distortion (RD) evaluation. The AIM-PB flag is signaled when AIM-PB is enabled in the sequence parameter set (SPS) and the current CU is an intra CU with a valid luma block in the luma coding tree. It is coded as a single regular context-adaptive binary arithmetic coding (CABAC) bin using a dedicated fixed context index of 0, without neighboring-CU-based context derivation. The flag is placed immediately after the

prediction-mode syntax. When the flag is one, the conventional luma intra prediction syntax, including the MIP, MRL, and ISP syntax elements, is skipped. Its estimated coding cost is included in the rate term during RD optimization.

4. Experimental Results and Analysis

4.1. Experimental Setup

AIM-PB was implemented in VTM 23.5 [25,26]. The VTM experiments followed the JVET CTC [27] including the test sequences, coding configurations, and numbers of coded frames. The All-Intra (AI) and Random-Access (RA) configurations were used at quantization parameter (QP) values of 22, 27, 32, and 37. Classes A1, A2, B, and C were tested under both configurations, while Class E was tested only under AI. Multiple independent coding jobs were executed concurrently throughout the VTM and NX2 experiments.

Additional experiments were conducted using NX2 [28] with test sequences and coding conditions specified in the Joint Call for Evidence (CfE) on video compression with capability beyond VVC [29]. The after-TIMD integration configuration was also evaluated using the JVET CTC sequences under AI to assess coding efficiency and runtime. The evaluated test sets and detailed NX2 results are presented in Section 4.5.

The VTM and NX2 experiments were conducted across five servers running Ubuntu 20.04.6 LTS. Two representative servers were equipped with Intel Core Ultra 9 285K processors (Intel Corporation, Santa Clara, CA, USA), with 256 GB and 128 GB RAM, respectively. For each sequence class or test set, the anchor and AIM-PB implementations were evaluated on the same server under the same build and execution conditions.

For both VTM and NX2, coding efficiency was evaluated using the Bjøntegaard delta rate (BD-rate) metric [30]. BD-rate results are reported relative to the corresponding anchor without AIM-PB. Negative values indicate average bit-rate savings at the same objective quality. EncT and DecT denote the encoding and decoding times, respectively, expressed as percentages of the corresponding anchor times.

4.2. Coding Performance on VTM

4.2.1. Coding Efficiency

Table 2 presents the coding performance and relative encoding and decoding times of AIM-PB with respect to the VTM 23.5 anchor. AIM-PB achieves average luma BD-rate reductions of 0.31% and 0.15% under the AI and RA configurations, respectively. The corresponding average Cb and Cr BD-rate reductions are 0.18% and 0.19% under AI and 0.17% and 0.16% under RA.

Luma BD-rate reductions are obtained for all sequences tested in each configuration. Under AI, the sequence-level luma BD-rate reductions range from 0.11% to 0.60%. The largest reduction of 0.60% is obtained for *FoodMarket4*, followed by *BasketballDrive* and *BasketballDrill* with reductions of 0.58% and 0.51%, respectively. Under RA, the luma BD-rate reductions range from 0.01% to 0.29%. AIM-PB selection behavior and predictor composition are further analyzed in Section 4.3.

4.2.2. Runtime

Under AI, the average encoding and decoding time ratios relative to the VTM 23.5 anchor are 109% and 100%, respectively. AIM-PB is evaluated as an additional luma intra prediction candidate through the existing full RD evaluation procedure. This evaluation involves not only prediction generation and blending but also residual coding processes, including transform, quantization, reconstruction, and CABAC-based bit estimation. Therefore, the additional encoding complexity is not limited to neighboring IPM collection and

blending-weight derivation. At the decoder, AIM-PB prediction is generated only for CUs coded using AIM-PB, without performing RD evaluation of competing prediction modes. Under RA, both average time ratios are 100%.

Table 2. BD-rate results and relative runtime of AIM-PB compared with the VTM 23.5 anchor.

Class	Sequence	All-Intra					Random-Access				
		Y	Cb	Cr	EncT	DecT	Y	Cb	Cr	EncT	DecT
A1	Tango2	−0.38%	−0.31%	−0.29%	115%	98%	−0.14%	−0.20%	0.01%	96%	100%
	FoodMarket4	−0.60%	−0.50%	−0.37%	114%	99%	−0.21%	−0.20%	−0.23%	102%	100%
	Campfire	−0.18%	−0.02%	−0.41%	100%	99%	−0.29%	−0.13%	−0.17%	103%	100%
A1 class		−0.39%	−0.28%	−0.35%	109%	99%	−0.22%	−0.17%	−0.13%	100%	100%
A2	CatRobot1	−0.24%	−0.13%	−0.11%	101%	99%	−0.16%	−0.04%	−0.22%	101%	100%
	DaylightRoad2	−0.33%	−0.24%	−0.20%	118%	100%	−0.21%	−0.54%	−0.14%	101%	100%
	ParkRunning3	−0.29%	−0.23%	−0.22%	106%	99%	−0.07%	−0.02%	−0.03%	102%	100%
A2 class		−0.29%	−0.20%	−0.18%	108%	99%	−0.15%	−0.20%	−0.13%	102%	100%
B	MarketPlace	−0.16%	−0.03%	−0.05%	107%	99%	−0.01%	0.15%	−0.12%	103%	100%
	RitualDance	−0.28%	−0.03%	−0.15%	105%	99%	−0.17%	0.03%	−0.10%	100%	100%
	Cactus	−0.15%	−0.02%	0.08%	109%	100%	−0.11%	−0.38%	−0.39%	98%	100%
	BasketballDrive	−0.58%	−0.37%	−0.37%	112%	101%	−0.25%	−0.06%	−0.11%	106%	100%
	BQTerrace	−0.32%	−0.35%	−0.06%	119%	101%	−0.16%	−0.46%	0.35%	94%	100%
B class		−0.30%	−0.16%	−0.11%	110%	100%	−0.14%	−0.14%	−0.07%	100%	100%
C	BasketballDrill	−0.51%	−0.07%	−0.06%	106%	100%	−0.25%	−0.34%	−0.35%	96%	100%
	BQMall	−0.23%	−0.12%	−0.09%	109%	101%	−0.03%	−0.30%	−0.16%	99%	101%
	PartyScene	−0.11%	−0.18%	−0.12%	117%	101%	−0.05%	−0.20%	−0.16%	104%	101%
	RaceHorses	−0.19%	−0.04%	−0.11%	106%	99%	−0.08%	0.16%	−0.55%	96%	100%
C class		−0.26%	−0.10%	−0.09%	109%	101%	−0.11%	−0.17%	−0.31%	99%	101%
E	FourPeople	−0.32%	−0.25%	−0.13%	113%	100%			–		
	Johnny	−0.30%	−0.36%	−0.47%	104%	100%			–		
	KristenAndSara	−0.33%	0.09%	−0.25%	109%	101%			–		
E class		−0.32%	−0.17%	−0.28%	109%	100%			–		
Overall		−0.31%	−0.18%	−0.19%	109%	100%	−0.15%	−0.17%	−0.16%	100%	100%

4.3. Coding Behavior Analysis

4.3.1. AIM-PB Selection Characteristics

Table 3 reports the AIM-PB selection ratios for each sequence class and QP under the AI configuration. For each class and QP, the selection ratio is calculated as the total number of final luma intra CUs coded using AIM-PB across all sequences in the class, divided by the corresponding total number of final luma intra CUs. The class-level selection ratios range from approximately 10% to 22% across the tested QPs, indicating that AIM-PB is selected across a wide range of sequence classes and QPs.

To examine the relationship between AIM-PB usage and coding performance in more detail, Table 4 compares the mean selection ratio of each sequence with its corresponding AI luma BD-rate. The mean selection ratio reported in the table is calculated as the arithmetic mean of the selection ratios at QPs 22, 27, 32, and 37.

FoodMarket4 and *BasketballDrive* exhibit relatively high mean selection ratios of 21.21% and 23.27%, respectively, together with luma BD-rate reductions of 0.60% and 0.58%. In

contrast, *Cactus*, *MarketPlace*, and *PartyScene* show lower selection ratios and smaller luma coding gains. However, *ParkRunning3* has a mean selection ratio of 21.79% but a luma BD-rate reduction of 0.29%. These results indicate that the selection ratio alone does not explain the differences in sequence-level coding gains. The selection ratios characterize tool usage rather than the magnitude of the RD improvement for each selected CU.

Table 3. AIM-PB selection ratios across sequence classes and QPs under AI.

Class	QP 22	QP 27	QP 32	QP 37
A1	12.02%	17.37%	21.68%	19.89%
A2	12.86%	20.07%	19.67%	19.14%
B	12.03%	17.27%	17.20%	16.06%
C	15.14%	14.74%	14.03%	12.44%
E	14.27%	13.88%	12.78%	10.29%
Overall	13.26%	16.67%	17.07%	15.56%

Table 4. Sequence-level mean AIM-PB selection ratio and luma BD-rate under AI.

Class	Sequence	Selection Ratio (%)	Y BD-Rate (%)
A1	Tango2	18.88	−0.38
	FoodMarket4	21.21	−0.60
	Campfire	15.33	−0.18
A2	CatRobot1	14.21	−0.24
	DaylightRoad2	18.31	−0.33
	ParkRunning3	21.79	−0.29
B	MarketPlace	11.18	−0.16
	RitualDance	13.73	−0.28
	Cactus	9.16	−0.15
	BasketballDrive	23.27	−0.58
	BQTerrace	19.04	−0.32
C	BasketballDrill	17.79	−0.51
	BQMall	13.16	−0.23
	PartyScene	11.99	−0.11
	RaceHorses	14.57	−0.19
E	FourPeople	13.01	−0.32
	Johnny	12.41	−0.30
	KristenAndSara	12.83	−0.33

4.3.2. Collected and Unique Angular Modes

To examine the number of prediction blocks blended when AIM-PB is selected, the distributions of N and K are analyzed. As defined in Section 3, N denotes the number of positions providing valid angular IPMs, whereas K denotes the number of unique angular modes. Each AIM-PB prediction uses K angular prediction blocks and one Planar prediction block. Figure 2 shows the distributions for AIM-PB-selected CUs.

The statistics were collected from a total of 8,843,432 AIM-PB-selected luma CUs across Classes A1, A2, B, C, and E under AI at QPs 22, 27, 32, and 37. The mean values of N and K are 2.494 and 1.446, respectively. CUs with $K = 1$ and $K = 2$ account for 46.28% and 32.44% of the selected CUs, respectively. Thus, the most frequent cases blend two or three prediction blocks, including Planar prediction. The mean total number of prediction blocks is 2.446 per selected CU, compared with the maximum of six.

The difference between N and K results from collecting the same angular IPM at multiple adjacent positions. Repeated occurrences contribute to weight derivation without requiring duplicate prediction generation. Consequently, the number of angular prediction blocks is determined by K, the number of unique collected modes, rather than being fixed at five.

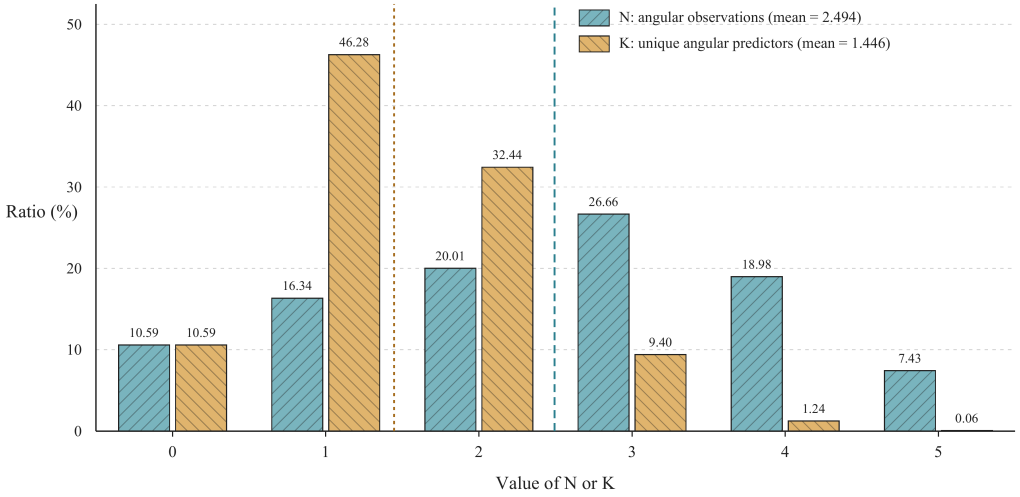


Figure 2. Distributions of N, the number of positions providing valid angular IPMs, and K, the number of unique angular modes, for AIM-PB-selected CUs under AI. The blue and orange dotted vertical lines indicate the mean values of N (2.494) and K (1.446), respectively.

4.3.3. Block-Size Characteristics

AIM-PB selection is analyzed in the Class B sequences under AI using three groups of luma CUs classified by block area. Small, medium, and large CUs are defined by

$$W \times H \leq 64, \qquad 64 < W \times H \leq 256, \qquad W \times H > 256,$$

respectively, where W and H denote the luma block width and height in samples. For each block-size group and QP, the selection ratio is calculated as the number of CUs coded using AIM-PB divided by the total number of CUs in that group. Table 5 reports the corresponding selection ratios for the three block-size groups across the tested QPs.

Table 5. AIM-PB selection ratios for different block-size groups in Class B under AI (%).

Block-Size Group	QP 22	QP 27	QP 32	QP 37
Small	12.37	17.48	16.94	16.11
Medium	11.40	17.32	17.47	15.89
Large	9.43	15.41	17.31	16.36

The AIM-PB selection ratios are broadly comparable across the three block-size groups. No single group consistently exhibits the highest selection ratio across the tested QPs, suggesting that AIM-PB does not consistently favor a particular block-size group within the tested Class B sequences.

4.3.4. Spatial Distribution of AIM-PB Selection

Figure 3 shows representative spatial distributions of AIM-PB-selected CUs under AI at QP 32.

To examine the spatial characteristics of AIM-PB selection, three sequences with different AI coding gains are analyzed: *FoodMarket4* , *ParkRunning3*, and *PartyScene*, whose

luma BD-rate reductions are 0.60%, 0.29%, and 0.11%, respectively. All visualizations use reconstructed frames coded at QP 32.

For each coded frame, the area-based AIM-PB selection ratio is calculated as the sum of the areas of AIM-PB-selected luma CUs divided by the total area of all luma CUs in that frame. For each sequence, the frame whose ratio is closest to the median of the per-frame ratios is selected for visualization. The selected representative frames show the spatial distribution of final luma CUs coded using AIM-PB.



Figure 3. Spatial distribution of luma CUs coded using AIM-PB in the representative frames under AI at QP 32: (**a**) FoodMarket4, POC 5; (**b**) ParkRunning3, POC 14; and (**c**) PartyScene, POC 18. Highlighted regions with yellow boxes indicate CUs coded using AIM-PB.

The area-based selection ratios are 20.64%, 20.40%, and 12.22% for *FoodMarket4*, *ParkRunning3*, and *PartyScene*, respectively. AIM-PB-selected CUs are observed at multiple locations within each representative frame. These visualizations supplement the sequence- and class-level selection statistics by showing the spatial distribution of selected CUs.

4.4. Design Analysis and Controlled Comparisons

Table 6 compares the coding performance of the proposed AIM-PB method and its design variants under the AI configuration. All BD-rate results are reported relative to the VTM 23.5 anchor.

4.4.1. Effect of Occurrence-Based Weighting

As shown in Table 6a, the equal-angular-weight configuration achieves a luma BD-rate reduction of 0.30%, compared with 0.31% for the proposed occurrence-proportional weighting. In this variant, the proposed Planar-weight schedule is retained, while the total angular weight is distributed equally among the K unique angular predictors. Thus, equal angular weighting retains nearly all of the luma coding gain, with an observed difference of only 0.01 percentage points. Based on the slightly larger measured coding gain, occurrence-proportional weighting is retained in the proposed AIM-PB design. This small difference does not establish statistical superiority of occurrence-proportional weighting.

Table 6. BD-rate performance of AIM-PB design configurations. (**a**) individual design variants and (**b**) equal-weight neighboring-position comparison.

Configuration	BD-Rate (%)		
	Y	Cb	Cr
(**a**) Individual design variants			
Proposed AIM-PB	−0.31	−0.18	−0.19
AIM-PB disabled for $N = 0$	−0.28	−0.16	−0.17
Without Planar; AIM-PB disabled for $N = 0$	−0.14	−0.09	−0.07
Equal angular weights	−0.30	−0.16	−0.20
(**b**) Equal-weight neighboring-position comparison			
Two-position configuration	−0.12	−0.02	−0.03
Three-position configuration	−0.23	−0.13	−0.12
Five-position configuration	−0.30	−0.14	−0.19
Nine-position configuration	−0.27	−0.14	−0.16

4.4.2. Contribution of Planar Prediction and the $N = 0$ Case

To examine whether the coding gain is mainly due to the Planar-only prediction used by AIM-PB for $N = 0$, a variant that disables AIM-PB for this case is evaluated. AIM-PB blending for $1 \leq N \leq 5$ is unchanged. As shown in Table 6a, this variant achieves a luma BD-rate reduction of 0.28%, compared with 0.31% for the proposed method. The retention of most of the gain indicates that the improvement is not primarily attributable to the Planar-only case at $N = 0$.

Table 6a also compares AIM-PB blending with and without Planar prediction, with AIM-PB disabled for $N = 0$ in both configurations. For $N \geq 1$, the configuration with Planar prediction combines angular and Planar prediction blocks, whereas the configuration without Planar prediction uses only angular prediction blocks. The luma BD-rate reductions are 0.28% with Planar prediction and 0.14% without Planar prediction. These results indicate that including Planar prediction makes a substantial contribution to the luma coding gain in the tested comparison.

4.4.3. Effect of the Neighboring-Position Configuration

To isolate the effect of the neighboring-position set, Table 6b compares the two-, three-, five-, and nine-position configurations under the same equal-weight condition. In this controlled comparison, Planar prediction and each angular predictor are assigned equal weights. The remaining AIM-PB processing and experimental settings are kept unchanged so that the comparison primarily reflects the effect of the examined neighboring-position set.

The two-, three-, five-, and nine-position configurations achieve luma BD-rate reductions of 0.12%, 0.23%, 0.30%, and 0.27%, respectively. The coding gain increases as the position set is expanded from two to five positions, whereas extending it further to nine positions does not provide an additional luma gain. Among the evaluated position sets under the common weighting condition, the five-position configuration provides the largest luma BD-rate reduction. This result supports the use of the five-position set in the proposed AIM-PB configuration for the tested coding conditions.

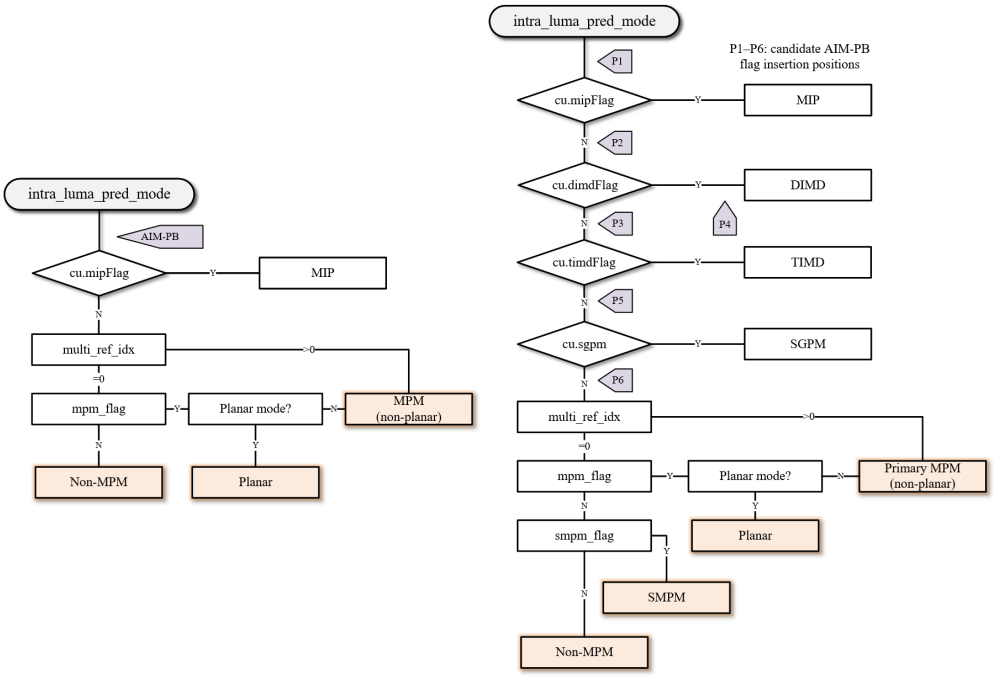
4.5. Evaluation in NX2

AIM-PB was also implemented in NX2 [28], which includes advanced intra prediction tools such as MIP, DIMD, TIMD, and spatial geometric partitioning mode (SGPM). This

evaluation examines whether AIM-PB provides additional coding gains in the presence of these tools.

4.5.1. Integration Configurations

Figure 4 illustrates simplified luma intra mode signaling flows in VTM 23.5 and NX2. In the VTM implementation, the AIM-PB flag is inserted immediately before the MIP flag. In the NX2 implementation, P1–P6 indicate the alternative AIM-PB flag insertion positions evaluated in this work.



(a) VTM 23.5 (b) NX2

Figure 4. Simplified luma intra mode signaling flows in (**a**) VTM 23.5 and (**b**) the tested NX2 implementation. The AIM-PB marker in (**a**) indicates the insertion position used in the VTM implementation. In (**b**), P1–P6 denote the alternative AIM-PB flag insertion positions evaluated separately. Tool-specific eligibility conditions and other syntax details are omitted for clarity.

In NX2, P1, P2, P3, P5, and P6 denote AIM-PB flag insertion positions before MIP, after MIP, after DIMD, after TIMD, and after SGPM, respectively. Each "after" position is located on the branch where the preceding tool is not selected. P4 represents the DIMD-submode configuration, in which an additional AIM-PB flag is signaled on the branch where DIMD is selected. Thus, P1–P6 represent alternative signaling configurations rather than sequential application or blending of AIM-PB with the corresponding tools.

Table 7 reports the coding performance of the six evaluated integration configurations.

Table 7. Overall NX2 BD-rate results for the evaluated AIM-PB integration configurations. P1–P6 correspond to the insertion positions in Figure 4.

Configuration	Y	Cb	Cr
Before MIP (P1)	−0.04%	0.04%	0.09%
After MIP (P2)	−0.05%	−0.04%	−0.04%
After DIMD (P3)	−0.05%	−0.11%	−0.03%
DIMD submode (P4)	−0.06%	−0.08%	0.10%
After TIMD (P5)	−0.06%	−0.04%	0.07%
After SGPM (P6)	−0.06%	0.02%	0.08%

All six evaluated integration configurations achieve luma BD-rate reductions, ranging from 0.04% to 0.06%. These improvements are obtained in the NX2 framework, where advanced intra prediction tools, including DIMD and TIMD, remain available. Thus, AIM-PB provides additional coding gains in the presence of these existing tools, rather than only when they are absent.

4.5.2. Performance Across Test Sequence Sets

The after-TIMD configuration (P5) is used as a representative configuration for the test-set analysis and the additional CTC evaluation. P4, P5, and P6 achieve the same reported luma BD-rate reduction of 0.06%, and P5 was selected as one representative configuration among these cases.

Table 8 shows luma BD-rate reductions ranging from 0.03% to 0.09% across the evaluated test sets, with an overall reduction of 0.06%.

Table 8. NX2 BD-rate results across test sets for the after-TIMD configuration.

Test Set	Y	Cb	Cr
SDR UHD/4K	−0.07%	0.03%	0.01%
SDR HD (RA)	−0.07%	−0.02%	0.00%
UGC HD	−0.03%	0.09%	0.25%
SDR HD (LB)	−0.09%	−0.25%	−0.02%
Gaming HD	−0.07%	−0.12%	0.04%
Overall	−0.06%	−0.04%	0.07%

The after-TIMD configuration was additionally evaluated under AI using sequence classes A1, A2, B, C, and E from the JVET CTC [27]. Coding efficiency and runtime were evaluated relative to the corresponding NX2 anchor without AIM-PB.

Table 9 reports the corresponding coding-efficiency and runtime results under the JVET CTC AI configuration.

Table 9. BD-rate and runtime results of AIM-PB with the after-TIMD integration (P5) in NX2 using CTC sequences under AI.

Class	Y	Cb	Cr	EncT	DecT
A1	−0.09%	0.14%	−0.10%	103%	99%
A2	−0.07%	0.01%	−0.06%	105%	100%
B	−0.07%	0.03%	0.03%	104%	100%
C	−0.03%	0.03%	−0.04%	100%	100%
E	−0.03%	0.13%	−0.13%	104%	100%
Overall	−0.06%	0.06%	−0.05%	103%	100%

As shown in Table 9, luma BD-rate reductions are obtained in all five evaluated classes, with an overall reduction of 0.06%. The reported encoding and decoding time ratios are 103% and 100%, respectively.

4.5.3. Decoder-Side Operation Analysis

To compare decoder-side computational complexity, operations were counted along the executed source paths of DIMD, normal TIMD, OBIC, and AIM-PB (P5) in the JVET-AN0267-based NX2 RA implementation. Six SDR HD sequences were evaluated at QPs 27, 32, 37, and 42, resulting in a total of 24 evaluated bitstreams. The RA configuration was selected because OBIC is disabled in the tested NX2 AI configuration but enabled together with DIMD and TIMD under RA, allowing DIMD, TIMD, OBIC, and AIM-PB to

be evaluated under the same software and coding environment. TIMD-SAD was disabled in this comparison.

The arithmetic-operation (AO) count is the unweighted sum of additions/subtractions, multiplications, and comparisons. Raw counts and corresponding denominators were accumulated across all bitstreams before normalization. Mode/weight derivation was normalized per derivation call, predictor counts per selected CU, and blending operations per selected luma sample. The analysis includes prerequisite DIMD/HoG processing for OBIC but excludes pre-selection neighbor collection and availability processing, common prediction-kernel arithmetic, auxiliary TIMD-SAD/SGPM processing, memory traffic, CABAC, residual reconstruction, and encoder RDO.

Because the reported quantities use different normalization units and correspond to tool-specific executed source paths, they are presented separately and are not combined into a single overall complexity score. Table 10 summarizes these normalized operation metrics under the stated counting rules and exclusions; they do not represent total decoder complexity or processor runtime.

Table 10. Decoder-side scalar-equivalent operation comparison in the tested JVET-AN0267-based NX2 RA environment.

Method	Add/Sub	Mult.	Comp.	AO/Call	Pred./CU	Blend AO/Sample
DIMD	756.75	194.92	1174.28	2125.96	5.93	36.93
TIMD	9242.67	65.99	1486.49	10795.14	2.25	6.38
OBIC	1156.25	290.36	2007.66	3454.27	5.94	32.84
AIM-PB (P5)	**9.11**	**1.54**	**27.95**	**38.59**	**2.54**	**8.08**

Bold indicates the proposed method.

As shown in Table 10, AIM-PB requires the fewest operations for mode and weight derivation, as it reuses neighboring IPMs without gradient analysis, template-cost evaluation, or HoC-based candidate ranking.

5. Discussion

AIM-PB derives angular prediction modes and blending weights directly from neighboring IPM information collected at five predefined adjacent positions, without additional processing such as gradient analysis or template-cost evaluation. The VTM results show luma BD-rate reductions for all tested sequences under both AI and RA, while the NX2 results show additional luma coding gains in the tested environment containing advanced intra prediction tools. The reported coding gains are evaluated relative to the corresponding VTM and NX2 anchors, rather than through a direct RD comparison with OBIC or weighted OBIC.

The coding behavior analysis shows that AIM-PB is selected across the evaluated sequence classes and QP values, with a separate analysis of block-size groups in Class B. The evaluated test sets cover various SDR resolutions and content types, including ultra-high definition (UHD)/4K, user-generated, and gaming video, providing an assessment of AIM-PB across different coding conditions.

The design comparisons in Section 4.4 show that including Planar prediction makes a substantial contribution to the luma coding gain when both compared configurations disable AIM-PB for $N = 0$. Equal angular weighting retains nearly all of the luma coding gain. Under the common equal-weight condition in Table 6b, the five-position configuration achieves the largest luma BD-rate reduction among the evaluated neighboring-position sets.

In terms of computational complexity, AIM-PB reuses existing neighboring IPM information from five predefined adjacent positions and generates one prediction block for

each unique angular mode, avoiding duplicate prediction generation. Under VTM AI, AIM-PB-selected CUs use an average of 1.446 angular predictors, or 2.446 predictors when Planar prediction is included. The per-sample weighted-sum computation is vectorized using single-instruction, multiple-data (SIMD) instructions.

Despite the additional prediction generation and blending operations, the measured results show improved luma coding efficiency with average decoding-time ratios of approximately 100% on the tested platforms, while requiring additional encoding time under AI. The NX2 framework already incorporates advanced intra prediction tools, including DIMD, TIMD, and SGPM, which may limit the opportunities for further coding-efficiency improvements. In this context, the additional 0.06% luma BD-rate reduction shows that AIM-PB can provide further coding gains even in the tested advanced intra prediction environment. The corresponding 3% increase in average encoding time represents a trade-off that should be considered according to the computational requirements of the target application.

6. Conclusions

This paper proposed AIM-PB, an intra prediction blending method that directly reuses angular IPMs from five predefined adjacent positions and combines the corresponding angular predictions with Planar prediction. The predictor modes and adaptive blending weights are derived from neighboring IPM information identically at the encoder and decoder, without requiring gradient analysis or template-cost evaluation.

In VTM 23.5, AIM-PB achieved average luma BD-rate reductions of 0.31% under AI and 0.15% under RA, with a maximum reduction of 0.60% for *FoodMarket4* under AI. Under the JVET CTC AI configuration, AIM-PB with after-TIMD integration (P5) in NX2 achieved an overall luma BD-rate reduction of 0.06%. The average decoding time ratios were approximately 100% relative to the corresponding anchors for VTM AI, VTM RA, and NX2 CTC AI. Analyses of selection behavior and predictor composition provided insight into how AIM-PB is used under the evaluated coding conditions, while controlled comparisons of key design choices provided further empirical support for the proposed blending design. Taken together, these results show that AIM-PB provides luma coding gains under the tested configurations, with average decoding-time ratios of approximately 100% on the tested platforms, while requiring additional encoding time under AI.

Author Contributions: Conceptualization, K.K.; methodology, K.K. and G.M.; software, K.K.; validation, K.K. and G.M.; formal analysis, K.K. and G.M.; investigation, K.K.; writing—original draft preparation, K.K.; writing—review and editing, G.M. and J.-G.K.; supervision, J.-G.K.; project administration, J.-G.K.; funding acquisition, J.-G.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) under Grant RS-2024-00356924, funded by the Ministry of Science and ICT (MSIT) of the Korean government.

Data Availability Statement: The experiments were conducted using the test sequences specified in the JVET CTC and the CfE on video compression with capability beyond VVC [27,29]. AIM-PB was implemented in VTM 23.5 and NX2 [26,28]. The experimental settings and evaluation procedures are described in the manuscript and the cited documents.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. ITU-T and ISO/IEC JTC 1. Versatile Video Coding. ITU-T Rec. H.266 and ISO/IEC 23090-3. 2020. Available online: https://www.itu.int/rec/T-REC-H.266-202008-S/en (accessed on 16 November 2026).

2. Bross, B.; Wang, Y.-K.; Ye, Y.; Liu, S.; Chen, J.; Sullivan, G.J.; Ohm, J.-R. Overview of the Versatile Video Coding (VVC) Standard and Its Applications. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 3736–3764. [CrossRef]
3. Ohm, J.-R.; Wien, M.; Bossen, F. Joint Call for Proposals on Video Compression with Capability Beyond VVC. Joint Video Experts Team of ITU-T and ISO/IEC, JVET-AQ2021. July 2026. Available online: https://jvet-experts.org/doc_end_user/documents/43_Geneva/wg11/JVET-AQ2021-v1.zip (accessed on 16 November 2026).
4. Pfaff, J.; Filippov, A.; Liu, S.; Zhao, X.; Chen, J.; De-Luxán-Hernández, S.; Wiegand, T.; Rufitskiy, V.; Ramasubramonian, A.K.; Van der Auwera, G. Intra Prediction and Mode Coding in VVC. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 3834–3847. [CrossRef]
5. Abdoli, M.; Guionnet, T.; Raulet, M.; Kulupana, G.; Blasi, S. Decoder-side intra mode derivation for next generation video coding. In Proceedings of the 2020 IEEE International Conference on Multimedia and Expo (ICME), London, UK, 6–10 July 2020; pp. 1–6. [CrossRef]
6. Blasi, S.; Lainema, J.; Zupancic, I.; Bugdayci Sansli, D. AHG12: Template-Based Intra Mode Derivation with Directional Blending. Joint Video Experts Team of ITU-T and ISO/IEC, JVET-AB0117. October 2022. Available online: https://jvet-experts.org/doc_end_user/documents/28_Mainz/wg11/JVET-AB0117-v3.zip (accessed on 16 November 2026).
7. Wang, H.; Seregin, V.; Karczewicz, M. EE2-1.11: Intra Prediction Fusion. Joint Video Experts Team of ITU-T and ISO/IEC, JVET-AB0148. October 2022. Available online: https://jvet-experts.org/doc_end_user/documents/28_Mainz/wg11/JVET-AB0148-v2.zip (accessed on 16 November 2026).
8. Youvalari, R.G.; Abdoli, M.; Tissier, A. EE2-2.2: Occurrence-Based Intra Coding (OBIC). Joint Video Experts Team of ITU-T and ISO/IEC, JVET-AH0076. April 2024. Available online: https://jvet-experts.org/doc_end_user/documents/34_Rennes/wg11/JVET-AH0076-v2.zip (accessed on 16 November 2026).
9. Chang, Y.-J.; Jhu, H.-J.; Jiang, H.-Y.; Zhao, L.; Zhao, X.; Li, X.; Liu, S.; Bross, B.; Keydel, P.; Schwarz, H.; et al. Multiple reference line coding for most probable modes in intra prediction. In Proceedings of the 2019 Data Compression Conference (DCC), Snowbird, UT, USA, 26–29 March 2019; p. 559. [CrossRef]
10. Bross, B.; Keydel, P.; Schwarz, H.; Marpe, D.; Wiegand, T.; Zhao, L.; Zhao, X.; Li, X.; Liu, S.; Chang, Y.-J.; et al. CE3: Multiple Reference Line Intra Prediction (Test 1.1.1, 1.1.2, 1.1.3 and 1.1.4). Joint Video Experts Team of ITU-T and ISO/IEC, JVET-L0283. October 2018. Available online: https://jvet-experts.org/doc_end_user/documents/12_Macao/wg11/JVET-L0283-v2.zip (accessed on 16 November 2026).
11. De-Luxán-Hernández, S.; George, V.; Ma, J.; Nguyen, T.; Schwarz, H.; Marpe, D.; Wiegand, T. An intra subpartition coding mode for VVC. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019; pp. 1203–1207. [CrossRef]
12. De-Luxán-Hernández, S.; Schwarz, H.; Marpe, D.; Wiegand, T. CE3: Line-Based Intra Coding Mode (Tests 1.4.1, 1.4.2 and 1.4.3). Joint Video Experts Team of ITU-T and ISO/IEC, JVET-K0049. July 2018. Available online: https://jvet-experts.org/doc_end_user/documents/11_Ljubljana/wg11/JVET-K0049-v2.zip (accessed on 16 November 2026).
13. Schäfer, M.; Stallenberger, B.; Pfaff, J.; Helle, P.; Schwarz, H.; Marpe, D.; Wiegand, T. An affine-linear intra prediction with complexity constraints. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019; pp. 1089–1093. [CrossRef]
14. Pfaff, J.; Stallenberger, B.; Schäfer, M.; Merkle, P.; Helle, P.; Hinz, T.; Schwarz, H.; Marpe, D.; Wiegand, T. CE3: Affine Linear Weighted Intra Prediction (CE3-4.1, CE3-4.2). Joint Video Experts Team of ITU-T and ISO/IEC, JVET-N0217. March 2019. Available online: https://jvet-experts.org/doc_end_user/documents/14_Geneva/wg11/JVET-N0217-v3.zip (accessed on 16 November 2026).
15. Abdoli, M.; Mora, E.; Guionnet, T.; Raulet, M. Non-CE3: Decoder-Side Intra Mode Derivation with Prediction Fusion. Joint Video Experts Team of ITU-T and ISO/IEC, JVET-N0342. March 2019. Available online: https://jvet-experts.org/doc_end_user/documents/14_Geneva/wg11/JVET-N0342-v5.zip (accessed on 16 November 2026).
16. Cao, K.; Hu, N.; Seregin, V.; Karczewicz, M.; Wang, Y.; Zhang, K.; Zhang, L. EE2-Related: Fusion for Template-Based Intra Mode Derivation. Joint Video Experts Team of ITU-T and ISO/IEC, JVET-W0123. July 2021. Available online: https://jvet-experts.org/doc_end_user/documents/23_Teleconference/wg11/JVET-W0123-v2.zip (accessed on 16 November 2026).
17. Zhang, L.; Wang, F.; Yu, Y.; Yu, H.; Wang, D. EE2-1.11: Intra Template Matching Prediction Fusion. Joint Video Experts Team of ITU-T and ISO/IEC, JVET-AD0072. April 2023. Available online: https://jvet-experts.org/doc_end_user/documents/30_Antalya/wg11/JVET-AD0072-v2.zip (accessed on 16 November 2026).
18. Huo, J.-Y.; Du, H.-Q.; Zhang, H.-L.; Qiao, W.-H.; Ma, Y.-Z.; Yang, F.-Z. EE2-1.16: A Fusion Method of Intra Template Matching Prediction (Intra TMP). Joint Video Experts Team of ITU-T and ISO/IEC, JVET-AD0116. April 2023. Available online: https://jvet-experts.org/doc_end_user/documents/30_Antalya/wg11/JVET-AD0116-v2.zip (accessed on 16 November 2026).
19. Andrivon, P.; Blestel, M. EE2-1.20: TIMD Fusion with Non-Angular Predictor. Joint Video Experts Team of ITU-T and ISO/IEC, JVET-AG0092. January 2024. Available online: https://jvet-experts.org/doc_end_user/documents/33_Teleconference/wg11/JVET-AG0092-v2.zip (accessed on 16 November 2026).

20. Lin, Y.-H.; Teng, C.-Y.; Liang, K.-W.; Yang, Y.-C.; Naser, K.; Dumas, T.; François, E.; Le Léannec, F. EE2-1.1: TIMD Fusion with Neural Network Based Intra Prediction. Joint Video Experts Team of ITU-T and ISO/IEC, JVET-AM0077. June 2025. Available online: https://jvet-experts.org/doc_end_user/documents/39_Daejeon/wg11/JVET-AM0077-v1.zip (accessed on 16 November 2026).
21. Abdoli, M.; Youvalari, R.G.; Plowman, F.; Tissier, A. Merge Mode for Template-Based Intra Mode Derivation (TIMD) in ECM. In Proceedings of the 2025 IEEE International Conference on Multimedia and Expo (ICME), Nantes, France, 30 June–4 July 2025; pp. 1–6. [CrossRef]
22. Luo, D.; Xiong, S.; Ren, C.; Sheriff, R.E.; He, X. Fusion-based versatile video coding intra prediction algorithm with template matching and linear prediction. *Sensors* **2022**, *22*, 5977. [CrossRef]
23. Andrivon, P.; Blestel, M.; Zouidi, N. EE2-1.9: Weighted OBIC. Joint Video Experts Team of ITU-T and ISO/IEC, JVET-AK0056. January 2025. Available online: https://jvet-experts.org/doc_end_user/documents/37_Geneva/wg11/JVET-AK0056-v2.zip (accessed on 16 November 2026).
24. JVET. Enhanced Compression Model (ECM), Version 20.0. Available online: https://vcgit.hhi.fraunhofer.de/ecm/ECM/-/tree/ECM-20.0 (accessed on 16 November 2026).
25. Browne, A.; Ye, Y.; Kim, S.H. Algorithm Description for Versatile Video Coding and Test Model 23 (VTM 23). Joint Video Experts Team of ITU-T and ISO/IEC, JVET-AO2002. January 2026. Available online: https://jvet-experts.org/doc_end_user/documents/41_Teleconference/wg11/JVET-AO2002-v1.zip (accessed on 16 November 2026).
26. JVET. VVC Reference Software VTM-23.5. Available online: https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/-/tags/VTM-23.5 (accessed on 16 November 2026).
27. Bossen, F.; Boyce, J.; Suehring, K.; Li, X.; Seregin, V. VTM Common Test Conditions and Software Reference Configurations for SDR Video. Joint Video Experts Team of ITU-T and ISO/IEC, JVET-T2010. October 2020. Available online: https://jvet-experts.org/doc_end_user/documents/20_Teleconference/wg11/JVET-T2010-v2.zip (accessed on 16 November 2026).
28. Ahmad, W.; Andersson, K.; Damghanian, M.; Gritsenko, V.; Litwic, L.; Liu, D.; Pettersson, M.; Shchukin, V.; Sjöberg, R.; Stegmaier, N.; et al. Ericsson, Fraunhofer HHI and Nokia Responses to Joint CfE on Video Compression with Capability beyond VVC. Joint Video Experts Team of ITU-T and ISO/IEC, JVET-AN0267. October 2025. Available online: https://jvet-experts.org/doc_end_user/documents/40_Geneva/wg11/JVET-AN0267-v1.zip (accessed on 16 November 2026).
29. Ohm, J.-R.; Wien, M.; Bossen, F. Joint Call for Evidence on Video Compression with Capability Beyond VVC. Joint Video Experts Team of ITU-T and ISO/IEC, JVET-AM2026. July 2025. Available online: https://jvet-experts.org/doc_end_user/documents/39_Daejeon/wg11/JVET-AM2026-v2.zip (accessed on 16 November 2026).
30. Bjøntegaard, G. Calculation of Average PSNR Differences Between RD-Curves. ITU-T Video Coding Experts Group. VCEG-M33. April 2001. Available online: https://www.itu.int/wftp3/av-arch/video-site/0104_Aus/VCEG-M33.doc (accessed on 16 November 2026).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.