Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Full length article

Using curved angular intra-frame prediction to improve video coding efficiency[☆]

Ramon Fernandes [a,*], Gustavo Sanchez [b], Rodrigo Cataldo [c], Luciano Agostini [d], César Marcon [a]

[a] *Pontifícia Universidade Católica do Rio Grande do Sul, Avenida Ipiranga 6681, Porto Alegre 90619-900, Brazil*
[b] *Instituto Federal Farroupilha, RS-377— Km 27, Alegrete 97555-000, Brazil*
[c] *Lab-STICC, Rue Saint-Maudé, Lorient 56100, France*
[d] *Universidade Federal de Pelotas, Rua Gomes Carneiro 01, Pelotas 96010-610, Brazil*

ARTICLE INFO

Keywords:
Intra-frame prediction
High Efficiency Video Coding (HEVC)
Video coding
Predictive coding

ABSTRACT

This article presents a new curved-based intra-frame prediction method for current and upcoming video coding standards. Our proposal extends conventional straight-line angular modes found on intra-prediction tools to model curved texture characteristics, enhancing the intra-frame prediction process. Our work targets the High Efficiency Video Coding (HEVC) standard for evaluation, although our curved-based method can be used by any other video coding standard. We model curved intra-frame prediction using an offset-based displacement calculation to each predicted sample. The proposal incurs a small bitstream overhead for transmitting the displacement information, which is offset by encoding efficiency gains. Experimental results demonstrate reduced residual energy; consequently, improving BD-Rate for the tested sequences. Evaluations applying eight curve displacement values show an average BD-Rate reduction of 2.69%, 2.49%, and 0.86% for All-Intra-8, All-Intra 10, and Random-Access configurations, respectively. The proposal allows further BD-Rate improvements, albeit at higher encoding complexity.

1. Introduction

The current digital video applications demand high quality and high definition [1]. Fast encoding for live-streaming, high compression ratios with minimal perceivable quality degradation, among others, are the goals commonly found in the state-of-the-art of video coding. Following these criteria, modern video encoders like the High Efficiency Video Coding (HEVC) [2], VP9 [3], the recent AV1 [4], and the upcoming Versatile Video Coding (VVC) [5] adopt advanced data structures [6,7] to evaluate different block sizes, considering the trade-off between encoding quality, bitrate, and computational complexity.

These encoders employ the hybrid encoding of prediction and residuals [8]. The predictive step adopts intra- and inter-frame prediction for encoding video frames. Intra-frame prediction explores the spatial redundancy of neighboring blocks [9], while inter-frame prediction is based on redundant information of past-encoded frames [10]. Once the predictive step is completed, the encoder transforms and quantizes the prediction error (referred to as the prediction residuals) [11]. Lastly, the quantized residuals and the high-level syntax elements of the encoding process are entropy coded and packed into the video bitstream.

The prediction process plays a crucial role in the encoded video quality and efficiency, as a high-quality prediction reduces the residual energy; consequently, reducing the bitstream size. Thus, the investigation of efficient prediction methods becomes essential for the current and next generations of video coding standards.

This article proposes new angular modes based on curved lines to be used in the intra-frame prediction of current and future video encoders. The current encoders use only straight lines in intra-frame prediction, and the use of curved lines enables capturing texture characteristics more accurately. This approach increases coding efficiency by reducing the prediction residuals, however, at an increased computational cost. Even so, increasing the encoding complexity is often acceptable since most broadcast video content is only encoded once but decoded countless times. This idea was proposed in our previous work [12], where preliminary results were presented, extending the prediction capabilities of the HEVC encoder.

To extend our original research, the main contributions of this work are:

[☆] This paper has been recommended for acceptance by Zicheng Liu.
[*] Corresponding author.
E-mail addresses: ramon.fernandes@acad.pucrs.br (R. Fernandes), gustavo.sanchez@iffarroupilha.edu.br (G. Sanchez), rodrigo.cataldo@acad.pucrs.br (R. Cataldo), agostini@inf.ufpel.edu.br (L. Agostini), cesar.marcon@pucrs.br (C. Marcon).

https://doi.org/10.1016/j.jvcir.2021.103291
Received 1 October 2020; Received in revised form 12 August 2021; Accepted 22 August 2021
Available online 29 August 2021

- A model to use curved prediction modes in the intra-frame prediction of current or future video encoders;
- A bitstream encoding method to reduce the overhead for signaling the use of curved prediction modes; and
- An in-depth exploration and analysis of our experimental results for several encoding configurations.

New intra-prediction methods have been proposed as enhancement extensions of HEVC, such as 3D-HEVC [13] and HEVC Screen Content (HEVC-SCC) [14]. Meanwhile, the recently released VVC adopts novel technologies for intra-prediction, like increasing angular intra-prediction modes [15], Multi-Reference Line (MRL) prediction [16], Position Dependent Prediction Combination (PDPC) [17] and Wide Angular Intra-Prediction (WAIP) [18]. The enhancements provided by these new technologies enable further coding efficiency gains for future video coding standards.

Moreover, at this point it is important to highlight that the HEVC was used as the baseline for our experimental results; thus, the remainder of this paper focuses on this video coding standard. Still, we emphasize that our method is not restricted to HEVC. The proposed intra-frame prediction method is applicable to all hybrid video standards, since the curved prediction modes extend the angular intra-frame prediction modes employed by these standards.

The remaining of this article is organized as follows. Section 2 presents an overview of the state-of-the-art regarding intra-frame prediction. Section 3 explains the HEVC intra-prediction structure. Details of the proposed method are shown in Section 4. Section 5 displays and discusses the experimental results, and Section 6 renders the conclusion of this work.

2. Related work

Most research addresses the computational complexity of the intra-frame predictors [19–24] adopted by modern coding standards, employing a multitude of optimization techniques. Meanwhile, fewer works address the intra-frame prediction process itself. The following works detail coding efficiency gains in the HEVC standard for the All-Intra coding configurations defined in the Common Test Conditions (CTC) [25].

Lucas et al. [26] propose a prediction algorithm for HEVC based on linear equations called Sparse Least Squares Prediction (SLSP), serving as an additional directional intra-prediction mode for complex texture areas. Results illustrate Peak Signal-to-Noise Ratio (PSNR) gains of up to 1 dB across all test sequences, with a bitrate reduction of at most 15%, at the cost of increasing an order of magnitude the encoding time. However, the test sequences used in this work do not follow the CTC.

Matsuo et al. [27] improve the generation of reference samples for intra-frame prediction. Their proposal uses a Discrete Cosine Transform Interpolation Filter (DCT-IF) to boost the intra-frame prediction performance. HEVC employs 2-sample interpolation to compute samples in intra-angular modes; i.e., a reference sample R_0 is used with its neighbor sample R_1. DCT-IF may use two or more reference samples, e.g., R_{-1}, R_0, R_1, R_2. Results demonstrate average coding gains from 0.31% to 0.34%, with maximum gains of 2.2% and an average increase of 102% in the encoding time.

Chen et al. [28] introduce constrained quantization for intra-prediction coding [29]. Samples of the intra-predicted blocks are encoded based on their location; samples far from the references are quantized and encoded, while an interpolation technique reconstructs the remaining samples. The experiments demonstrate a Bjøntegaard Delta Rate (BD-Rate) [30] reduction of 2% with an increase of 130% in the encoding time.

Usually, intra-frame predictors analyze one line on the edge of previously decoded blocks to extrapolate block contents. In [16,31], MRL extends this model to consider up to four lines of samples into the neighboring block edge. The encoder chooses which reference line to use according to standard rate-distortion metrics, signaling the used reference line in the video bitstream. Results demonstrate an average BD-Rate reduction of 2.3%, with a maximum of 4.3%; however, at a nearly four times increase in computational complexity.

The related work demonstrates that exploiting intra-frame prediction algorithms improves video coding efficiency due to the better prediction of the encoded blocks. The proposal of using curved lines for intra-frame prediction is a novel and promising approach for enhancing the prediction process, as it can model characteristics in natural images with higher accuracy than the usual straight-lines adopted in video coding standards.

3. HEVC intra-frame prediction

This section details the HEVC intra-frame predictor since the proposed prediction method is evaluated inside the HEVC encoder. The tools for intra-frame prediction in HEVC exploit the inherent spatial redundancy inside a video frame to compress information, using the high correlation of nearby samples and blocks [32][33]. HEVC employs a range of prediction methods to predict different kinds of content efficiently. These methods [9] can be classified in:

1. angular prediction, to model directional features present in images; and
2. planar and DC prediction, designed to model smooth image content.

The HEVC intra-frame prediction uses reference samples from adjacent (above and left) reconstructed blocks to extrapolate the predicted block contents. Fig. 1 displays the 33 directions of HEVC angular intra-frame prediction along with their angular parameter A, representing the direction of each mode. Modes 2 through 17 correspond to horizontal modes, while modes 18 through 34 correspond to vertical modes. Although angular prediction is suitable for approximating blocks with highly contrasting edges, it may result in blockiness or visible contouring in smooth image regions [9]. Planar prediction mode overcomes these issues by creating a prediction surface without discontinuities at the block boundaries. Meanwhile, DC prediction is efficient for image regions with homogeneous textures, populating the predicted block with an average constant value computed from the reference samples.

The intra-frame prediction employs a reference sample array obtained from neighboring reconstructed blocks with $2n+1$ elements in the range $[-1, 2n-1]$, where n corresponds to the predicted block dimension, measured as $n \times n$. The values for the reference array are obtained by applying Eq. (1) and Eq. (2) [34] for horizontal and vertical modes, respectively. Fig. 2 presents a visual representation of the reference array.

$$ref[x] = p[-1+x][-1], \ (x \geq 0) \tag{1}$$

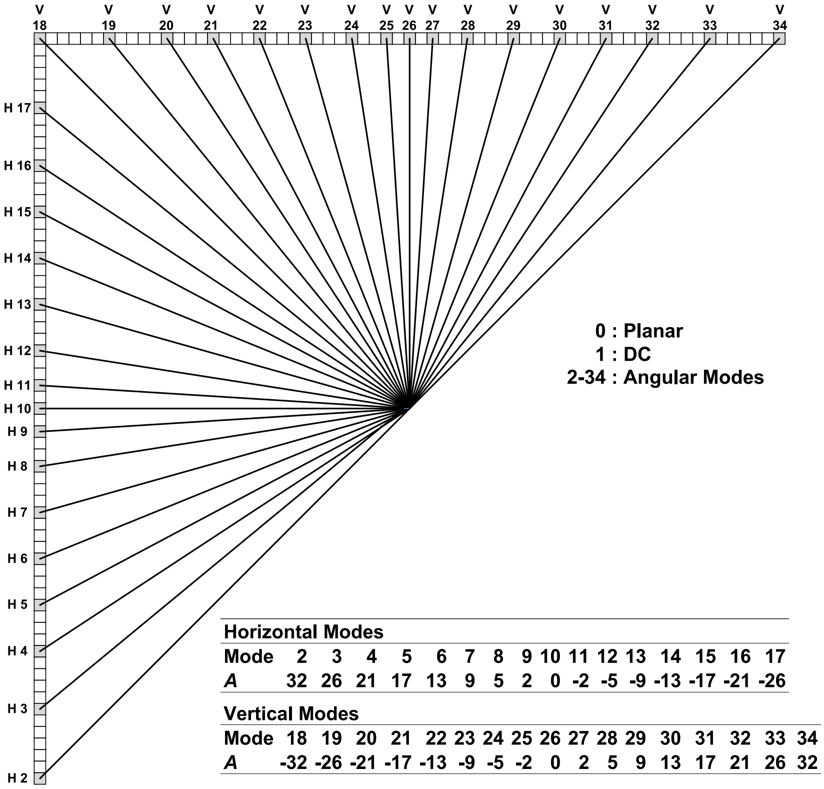
$$ref[y] = p[-1][-1+y], \ (y \geq 0) \tag{2}$$

For prediction modes with negative angular parameters, the reference array range is $[-n+1, n-1]$, where a process called *reference array extension* computes values in the range $[-n+1, -1]$. Eq. (3) and Eq. (4) show a projection from the left and above reference arrays used to obtain these values for vertical and horizontal modes, respectively; B represents the inverse angle parameter, which depends on the angular prediction mode.

$$ref[x] = p[-1][-1+((x \times B \times 128) \gg 8)], \ (x < 0) \tag{3}$$

$$ref[y] = p[-1+((y \times B \times 128) \gg 8)][-1], \ (y < 0) \tag{4}$$

The reference software implementation of HEVC (HTM) evaluates recursively evaluates all levels inside a quadtree structure to select the best intra-frame prediction mode, with block size ranging from 64×64



Horizontal Modes

Mode	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
A	32	26	21	17	13	9	5	2	0	-2	-5	-9	-13	-17	-21	-26

Vertical Modes

Mode	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
A	-32	-26	-21	-17	-13	-9	-5	-2	0	2	5	9	13	17	21	26	32

Fig. 1. Graphical representation of the HEVC intra-frame prediction modes.

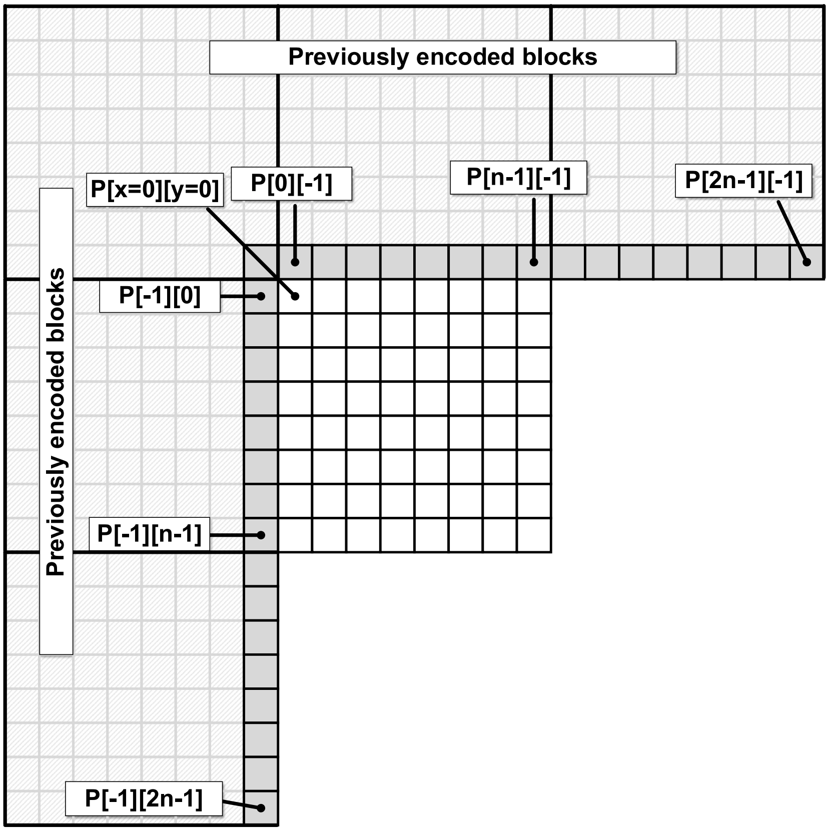


Fig. 2. HEVC intra-frame prediction top and left reference sample arrays. The intra-predictor uses samples from the edge of previously encoded blocks to extrapolate the predicted block contents.

down to 4×4. The quadtree subdivision depends on the block contents and encoding parameters regarding the desired coding efficiency. Each leaf node of the quadtree encodes information like the prediction mode into a Prediction Unit (PU) and transformed residual information into a Transform Unit (TU). HTM evaluates, for each block size, all 35 intra-frame prediction modes employing Rough Mode Decision (RMD) [35]. This evaluation uses the Sum of Absolute Transformed Differences (SATD) to roughly estimate the encoding efficiency without the need to compute the complete Rate-Distortion (RD) cost [32]. The most efficient modes, along with the Most Probable Modes (MPM) [36], are inserted into an RD-list to be later fully evaluated. The MPM list is populated according to the prediction modes of neighboring blocks.

4. Curve-based intra-frame prediction

The core contribution of this work is a novel approach to compute intra-frame predicted blocks. Each sample in a predicted block is subject to a displacement operation, based on the sample position together with a curve-offset. The idea is to offer a refined prediction model that can represent additional texture characteristics. Our model was inserted in the HEVC reference software for evaluation, extending the HEVC intra-frame predictor with additional modes.

Eqs. (5)–(10) [34] demonstrate how HEVC computes samples for vertical and horizontal angular prediction modes. Eq. (5) and Eq. (8) perform linear interpolation using the two closest reference samples in the prediction direction to compute samples p for vertical and horizontal modes, respectively. Eq. (6) and Eq. (9) show the projected integer displacement i depends on the row y and column x for vertical and horizontal modes, respectively, and the angular parameter A (shown in Fig. 1). Finally, the fractional part of the projected displacement f (Eq. (7) and Eq. (10)) depends on the row y and column x for vertical and horizontal modes, respectively, as well as the angular parameter A.

$$p[x][y] = ((32 - f) \times ref[x + i + 1] + f \times ref[x + i + 2] + 16) \gg 5 \tag{5}$$

$$i = ((y + 1) \times A) \gg 5, \text{ for vertical modes} \tag{6}$$

$$f = ((y + 1) \times A)\&31, \text{ for vertical modes} \tag{7}$$

$$p[x][y] = ((32 - f) \times ref[y + i + 1] + f \times ref[y + i + 2] + 16) \gg 5 \tag{8}$$

$$i = ((x + 1) \times A) \gg 5, \text{ for horizontal modes} \tag{9}$$

$$f = ((x + 1) \times A)\&31, \text{ for horizontal modes} \tag{10}$$

We model the curve parameter according to the intra-frame prediction direction, applying a displacement according to the sample position in the predicted block. The curves represent an arc extending away from the prediction direction, either to the left (negative displacement) or to the right (positive displacement). We refer to this displacement value as ω. Considering the idea of fitting a curve parameter in the HEVC intra-frame interpolator, let Hb be half of the block size, d be the distance from the predicted sample to the block centerline, given by Eq. (11) and Eq. (12) for vertical and horizontal modes, respectively. Then, Eq. (13) describes the displacement applied to the reference sample computed by HEVC intra-frame prediction. Eq. (13) demonstrates that samples located near the border of a prediction block are less affected by ω, while samples located near the center are most affected by the curve offset weight since $Hb - d$ is minimum in the borders and maximum in the center of the block. Fig. 3 shows this behavior for an 8×8 prediction block.

$$d = \begin{cases} Hb - y - 1, & y < Hb \\ y - Hb, & y \geq Hb \end{cases} \tag{11}$$

$$d = \begin{cases} Hb - x - 1, & x < Hb \\ x - Hb, & x \geq Hb \end{cases} \tag{12}$$

$$shift = (Hb - d) \times \omega / Hb \tag{13}$$

By solving Eq. (13), the $shift$ parameter is linked to Eq. (5) and Eq. (8) at the reference sample-indexing step, resulting in Eq. (14) and Eq. (15), for vertical and horizontal modes, respectively; these equations are then used to compute reference sample values in predicted blocks. Note that when $\omega = 0$, i.e., without a curve-offset weight, the prediction behaves exactly like in the standard HEVC. In either case, computing the angular intra-frame predicted samples requires low additional computational effort, as the only required step is solving Eq. (13) and the modified Eq. (14) and Eq. (15), as well as the reference array extension, described as follows.

$$\begin{aligned} p[x][y] = {} & ((32 - f) \times ref[x + i + shift + 1] \\ & + f \times ref[x + i + shift + 2] + 16) \gg 5 \end{aligned} \tag{14}$$

$$\begin{aligned} p[x][y] = {} & ((32 - f) \times ref[y + i + shift + 1] \\ & + f \times ref[y + i + shift + 2] + 16) \gg 5 \end{aligned} \tag{15}$$

4.1. Reference array extension

The displacements can cause some predicted samples to index reference samples beyond the standard reference array range shown in Fig. 2. Therefore, Fig. 4 illustrates that our model always requires the computation of the *reference array extension*, as well as a wider reference array with $3n$ elements, where n corresponds to the predicted block size. As in HEVC, the *reference array extension* is used to compute reference samples indexed in the range of $[-n, -1]$; the remaining reference samples in the range of $[0, 2n-1]$ are generated using the same method employed by HEVC. The choice of $3n$ samples considers the full range of the default reference array in HEVC in addition to its extended range. With prediction modes using a positive angular parameter, HEVC defines reference array indexes in the range $[-1, 2n-1]$, thus $2n+1$ samples. Meanwhile, for prediction modes using a negative angular parameter, HEVC defines reference sample array indexes in the range $[-n, n - 1]$, thus $2n$ samples. Overlapping both reference array ranges corresponds to the reference array range $[-n, 2n - 1]$, thus $3n$ samples.

In large prediction blocks or when ω applies a large displacement, the predicted samples computed by Eq. (14) and Eq. (15) may index elements outside the range $[-n, 2n-1]$. In these situations, we truncate the reference array index to ensure that it points inside the allowed range of $[-n, 2n - 1]$.

4.2. Syntax elements and bitstream considerations

HEVC defines four syntax elements for an intra-predicted PU, namely:

1. `prev_intra_luma_pred_flag`, a 1-bit flag, derived from neighboring blocks, that specifies if a PU is predicted with MPM;
2. `mpm_idx`, a 2-bit long element, which indicates the MPM index used;
3. `rem_intra_luma_pred_mode`, a 5-bit long element present when MPM is not used, which identifies the intra-prediction mode for the PU; and
4. `intra_chroma_pred_mode`, a 3-bit long element, which identifies the chroma prediction mode in the PU.

Our proposal requires one additional syntax element, `curve_offset`, to signal the ω value used by the PU. The length of this element depends on the number of curve-offsets used, called θ, which is always an even number to achieve symmetric displacements, such that $\theta = 2$ represents $\omega = \{-1; 1\}$, $\theta = 4$ is $\omega = \{-1, 1, -2, 2\}$, and so on. We also use the neighboring block information of MPM for identifying the prediction mode; however, it is still necessary to transmit the ω value of each PU.

The bitstream representation for `curve_offset` adopts static Huffman codes. To avoid any biased results, we derive the Huffman codes according to a separate group of test sequences, shown in Table 1.

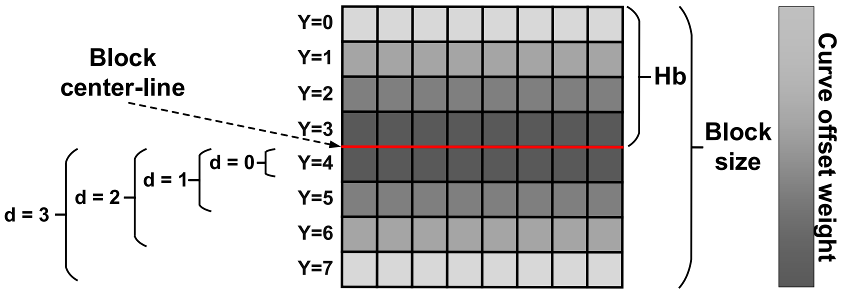


Fig. 3. Curve offset weights (ω) for an 8 × 8 prediction block in vertical mode. Samples located near the center-line are more influenced by the ω, while samples further away from the center-line are less affected [12].

Table 1
Training sequences for ω usage estimation.

Resolution (pixels)	Sequence name	Frame count	Frame rate (fps)	Bit depth
3840 × 2160	Foreman	515	60	10
3840 × 2160	Mobile	738	60	10
2560 × 1600	NebutaFestival	301	60	10
2560 × 1600	PeopleOnStreet	150	60	10
1920 × 1080	ArenaOfValor	600	60	8
1920 × 1080	Kimono	240	24	8
1920 × 1080	HoneyBee	600	120	8
1024 × 768	ChinaSpeed	500	30	8

These test sequences provide a wide range of resolutions and texture characteristics found in encoded videos.

Our approach for signaling `curve_offset` always requires at least one additional bit to be present. Results for the training sequence have demonstrated that $\omega = 0$, i.e., planar, DC, and angular modes represented by a straight-line, are most frequently used. Thus, the Huffman tree for encoding `curve_offset` encodes $\omega = 0$ with a single bit, while the remaining ω values require at least two bits. An example of the Huffman tree structure is shown in Fig. 5. As the θ value restricts the number of available curve-offsets, there exists a different Huffman tree for each value of θ adopted in our experiments. Table 2 illustrates all Huffman codes used in our experiments, obtained while encoding the training sequences shown in Table 1, where the longest binary length for encoding ω is 8 bits, which occur for $\theta = 30$ and $\theta = 32$.

5. Setup & results

The set of tests employed the HEVC test Model (HM) version 16.18 [37] with All-Intra and Random-Access, using 8-bit and 10-bit depths for both configurations. We have tested each configuration with all test sequences detailed in Table 3, according to CTC version AF1100 [25]. Since the core contribution of our work is in the intra-prediction tools, the results illustrate All-Intra-8 and All-Intra-10 configurations separately, while Random-Access corresponds to the average gains of its 8- and 10-bit configurations.

Table 2
Huffman codes used for different values of θ. The codeword length depends on the frequency of occurrence of each value of ω for encoding the training sequences.

θ		2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32
	−16	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	111111
	−15	–	–	–	–	–	–	–	–	–	–	–	–	–	–	10000	1111010
	−14	–	–	–	–	–	–	–	–	–	–	–	–	–	10001	1111101	1100110
	−13	–	–	–	–	–	–	–	–	–	–	–	–	10111	100110	1111010	1110000
	−12	–	–	–	–	–	–	–	–	–	–	–	11010	100101	1111100	1110010	1100111
	−11	–	–	–	–	–	–	–	–	–	–	11010	100001	1110100	1101001	1100110	1100100
	−10	–	–	–	–	–	–	–	–	–	11011	101111	1111111	1110101	1101010	1100111	1100101
	−9	–	–	–	–	–	–	–	–	11110	111001	101101	100010	1111110	1111101	1111011	1111011
	−8	–	–	–	–	–	–	–	11111	110111	101110	100011	100011	100001	1111110	1111111	1111101
	−7	–	–	–	–	–	–	1000	10001	110101	101111	101110	100111	100100	100111	100011	100110
	−6	–	–	–	–	–	1010	10100	111011	111000	110101	110110	101111	110100	101110	110001	100111
	−5	–	–	–	–	1011	10011	111011	111010	110110	111000	110111	110110	110101	101111	110010	110001
	−4	–	–	–	1110	11110	10110	10011	10100	10010	10000	111111	111110	111110	111100	111100	111010
	−3	–	–	101	1001	11111	11110	11100	11001	10101	10110	10011	10110	10011	10110	10011	10010
	−2	–	111	1101	1011	1010	1000	11111	11110	11111	11110	11110	11101	11100	11100	11011	11011
	−1	11	100	1111	1101	1101	1100	1100	1011	1011	1010	1010	1010	1010	1010	1010	1010
ω	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	10	101	100	1111	1110	1110	1101	1101	1100	1100	1100	1100	1100	1100	1011	1011
	2	–	110	1100	1010	1001	11111	11110	11100	11101	11101	11100	11100	11011	11011	11010	11010
	3	–	–	1110	1000	11001	11010	10111	10101	10100	10011	10010	10010	10001	10010	10010	10001
	4	–	–	–	1100	11000	10010	10010	10000	10000	111110	111011	111100	111011	111010	111010	111001
	5	–	–	–	–	1000	10111	10101	10011	10001	111111	111110	111101	111101	111011	111011	111100
	6	–	–	–	–	–	11011	111010	110001	110100	100101	100010	100000	100000	1111111	100010	100000
	7	–	–	–	–	–	–	10110	110000	100110	100011	100000	1111110	1111000	1101011	1110001	1110110
	8	–	–	–	–	–	–	–	10010	100111	100100	100001	1101111	1111001	1111010	1110011	1110001
	9	–	–	–	–	–	–	–	–	111001	100010	1110100	1101110	1011011	1101000	1100001	1100000
	10	–	–	–	–	–	–	–	–	–	110100	1110101	1011101	1011001	1000010	1100000	1000011
	11	–	–	–	–	–	–	–	–	–	–	101100	1011100	1011000	1000000	11111001	11111001
	12	–	–	–	–	–	–	–	–	–	–	–	100110	1011010	1000001	11111100	1000010
	13	–	–	–	–	–	–	–	–	–	–	–	–	1111111	1000011	11111000	11101111
	14	–	–	–	–	–	–	–	–	–	–	–	–	–	1111011	11111101	11101110
	15	–	–	–	–	–	–	–	–	–	–	–	–	–	–	1110000	11111000
	16	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	1100001
Average length		1.38	1.94	2.32	2.57	2.76	2.92	3.05	3.17	3.25	3.34	3.41	3.47	3.53	3.59	3.64	3.69

Table 3
HEVC AF1100 test sequences used for experimentation.

Class	Resolution	Sequence name	Frame count	Frame rate (fps)	Bit depth
Class A1	3840 × 2160	Tango2	294	60	10
		FoodMarket4	300	60	10
		Campfire	300	30	10
Class A2	3840 × 2160	CatRobot	300	60	10
		DaylightRoad2	300	60	10
		ParkRunning3	300	50	10
Class B	1920 × 1080	MarketPlace	600	60	10
		RitualDance	600	60	10
		Cactus	500	50	8
		BasketballDrive	500	50	8
		BQTerrace	600	60	8
Class C	832 × 480	BasketballDrill	500	50	8
		BQMall	600	60	8
		PartyScene	500	50	8
		RaceHorses	300	30	8
Class D	416 × 240	BasketballPass	500	50	8
		BQSquare	600	60	8
		BlowingBubbles	500	50	8
		RaceHorses	300	30	8
Class E	1280 × 720	FourPeople	600	60	8
		Johnny	600	60	8
		KristenAndSara	600	60	8
Class F	832 × 480	BasketballDrillText	500	50	8
	1920 × 1080	ArenaOfValor	600	60	8
	1280 × 720	SlideEditing	300	30	8
		SlideShow	500	20	8

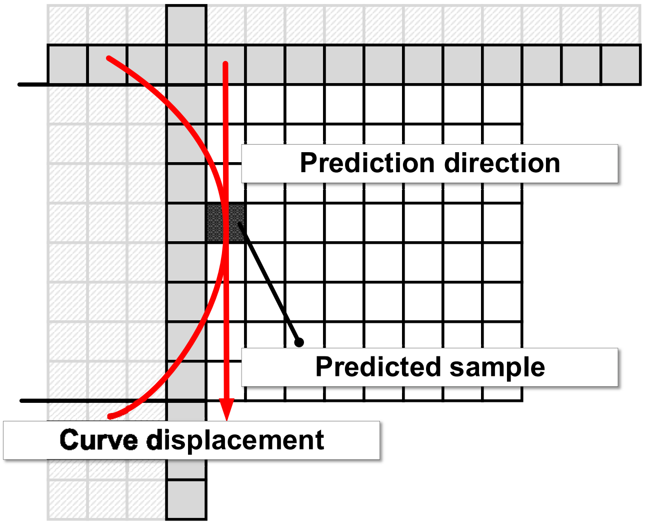


Fig. 4. Extended reference array for intra-frame prediction given the sample displacement for curves.

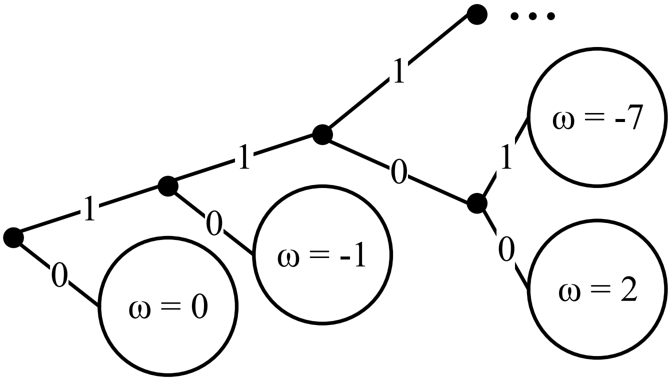


Fig. 5. Example of Huffman tree representation for signaling the ω parameter in the PU.

Even with almost all previously published works with solutions targeting intra-frame prediction presenting results considering only the All-Intra configuration, we also decided to evaluate the Random-Access configuration. The All-Intra configurations use only I-type frames, which are entirely encoded using intra-frame prediction; thus, this configuration is interesting to isolate the effects of any modification in the intra-prediction process. Otherwise, the Random-Access configuration allows the use of I-, P- and B-type frames where the P- and B-type frames use both intra- and inter-frame prediction, with inter-frame being used most of the time [34]. In this case, the improvements in intra-prediction are masked since it is less used. Considering both configurations in our experiments, we can guarantee a precise evaluation of the impacts at the intra-prediction when using the All-Intra configurations and a more realistic global evaluation when using the Random-Access configurations.

The experiments allow evaluating the proposed method by analyzing the (i) utilization of intra-prediction modes with curves; (ii) residual energy of the test sequences with standard HEVC encoding; (iii) coding efficiency with BD-Rate [30]; and (iv) computational overhead of our model. We have configured the encoder to consider only the Rate-Distortion Optimization (RDO) measurements, as this enabled a proper evaluation of our model regarding its prediction quality and bitrate costs.

The inclusion of curved lines creates a new set of prediction modes. For each value of ω, the encoder evaluates the 33 angular modes of HEVC with the corresponding displacement. This change quickly increases the number of prediction modes P, given as:

1. 2 for Planar and DC;
2. 33 for the angular modes; and
3. $\theta \times 33$ for curved modes.

Therefore, we define P as $P(\theta) = 2 + (33 \times (\theta + 1))$.

We have evaluated θ from 2 to 32, resulting in 101 up to 1025 evaluated intra-prediction modes. The choice of using θ in the range $[2, 32]$ considers the lower bound $\omega = -1$ and $\omega = 1$ ($\theta = 2$), which are the minimum curve displacement values. We have chosen the upper bound as $\theta = 32$ as it represents a reasonable cutoff point for the experiments, providing an extensive evaluation of several curve displacement values. This large number of tests enables an in-depth evaluation of our proposal in coding efficiency, as well as the saturation point analysis, where additional displacements no longer offer significant benefits.

5.1. Curved modes utilization

The intensity of the generated curve depends on ω; the higher its value, the more accentuated is the arc of the curve. Results revealed that low accentuated curves are more frequently used. This analysis considered only the All-Intra-8 configuration, avoiding the influence of inter-prediction present in Random-Access. Fig. 6 displays this characteristic where $\omega = 0$ are the default HEVC intra-predictor angular modes, and $\omega < 0$ and $\omega > 0$ correspond to curves displaced to the left and right of the prediction direction, respectively.

Despite angular modes with $\omega = 0$ having high utilization, curves are still significantly employed. Fig. 6 illustrates the distribution of prediction modes used by each sample encoded in the test sequences, for $\theta = 32$. This relation considers only angular modes, disregarding samples encoded with Planar and DC modes. With $\theta = 32$, 45.34% and 54.66% of the samples are non-curved and curved modes, respectively. For $\theta = 2$, 66 additional angular modes are available, and these modes are used to encode 25.95% of the samples.

The number of encoded samples for different θ values is shown in Fig. 7, considering the results for the Random-Access configuration and samples encoded with Planar and DC modes. The addition of angular modes with curves causes a reduction in the utilization of conventional HEVC intra-frame prediction modes, more prominently for the angular modes. Fig. 7 illustrates that the use of curved modes grows quickly for low values of θ, and when $\theta > 8$, the use of additional curves offers little additional benefits. It is important to emphasize that Planar and DC indicate individual modes, where Angular and Curved indicate a group of modes. Angular considers 33 modes, and Curved considers a variable number of modes, depending on θ, as previously discussed.

An interesting result shown in Fig. 8 is that the same conclusions presented in Section 5.1 are valid here: the most significant gains occur for lower values of θ. Again, the increase in the number of evaluated curves with $\theta > 8$ brings a smaller reduction in residual energy. For a value of $\theta = 8$, ω is in the range $[-4; 4]$, as discussed in Section 4.2, which is consistent with the modes with higher utilization shown in Figs. 6 and 7.

5.2. Coding efficiency

Table 4 details all experiments for All-Intra-8, All-Intra-10, and Random-Access configurations, varying θ from 2 up to 32 and grouping the video sequences in classes, as defined by the CTC. The overall average is also presented for each configuration and θ value. The BD-Rate measurement gives the encoding efficiency gains for each encoder implementation, where negative values, shown as a percentage, denote higher coding efficiency. In our experiments, each θ corresponds to a different encoder implementation, increasing the available curved modes for encoding the test sequences.

The results in Table 4 show that the proposed method reaches substantial coding efficiency gains for all video classes when considering the All-Intra-8 or the All-Intra-10 configurations. Indeed, these gains were found in all encoded sequences for these configurations. The experiments with the Random-Access configuration show that most results also presented coding efficiency gains, which tend to be lower than those found in the All-Intra-8 and All-Intra-10 configurations because intra-prediction is less used in the Random-Access configuration. The Random-Access results presented coding efficiency losses for some classes and θ configurations due to the extra bits used to signal the curved modes; the higher is the number of curved modes, the higher is the number of bits required to represent these modes.

Regarding the reduced coding performance gains for the Random-Access results, we have two hypotheses considering the Random-Access configuration:

1. Localized changes in the coding flow due to the curved prediction modes translate into different frame partitioning and used prediction modes. While these changes may improve the coding efficiency of an I-frame, the same may not hold true for P- and B-frames, as these last frames reference I-frames for prediction. This situation may also affect the next hypothesis.
2. The curved prediction modes, prominent in I-frames, may generate visual artifacts that affect inter-frame prediction efficiency. Since the HEVC inter-frame prediction tools were never envisioned to work with curved prediction modes, they may require adaptations to efficiently work with our proposal.

An important observation must be made regarding the Random-Access results of classes A1 and A2. Although these video classes show reduced coding efficiency with higher θ values, they achieve slightly better coding efficiency with $\theta = 2$, progressively losing these gains as θ increases. A similar situation occurs for video class B: although it achieves coding efficiency gains for all θ values, these gains decrease as θ increases similarly to classes A1 and A2. We conclude that the higher definition videos of these video classes are more susceptible to coding efficiency variations due to the hypotheses mentioned above that affect inter-frame prediction performance.

One interesting situation occurs for the Class F videos, which consist mostly of screen and text content; for all encoding configurations, curved modes have offered coding efficiency gains. For Random-Access, the Class F videos have consistently shown improved BD-Rate values. A reason for the performance of our method in this class of videos is due to the naturally curved textures within the text content, as Fig. 9 shows, where the blue blocks are encoded using curved modes. This behavior occurs throughout all frames of Class F videos in the presence of text content.

Similar to the results shown in Fig. 9, curved prediction blocks concentrate around the naturally occurring curves of textures like hair, tree branches, and clothing, shown in Fig. 10. The utilization of curved modes for these PUs provides better coding efficiency with lower residual energy since the curved prediction modes offer a more accurate approximation of textures in these situations.

Fig. 11 shows the average coding efficiency of all classes displayed in Table 4 for 16 values of θ and the three configurations. Fig. 11 allows us to infer that all implementations presented gains in coding efficiency when the average is considered, even when Random-Access is used. Again, as observed in the previous subsections, the most significant gains of BD-rate occur with lower values θ. In this case, with $\theta > 6$, the gains start to increase slowly. Random-Access configurations show even a reduction in the gains for higher values of θ, which is caused by the extra bits required to signalize the curved modes, as previously discussed.

5.3. Encoding and decoding complexity

The main drawback of increasing prediction modes is the higher computational cost. Fig. 12 illustrates the average encoding complexity for each encoder configuration. At $\theta = 2$, the encoding complexity increases to near thrice the reference value of the HTM for the All-Intra configurations. This behavior continues as the amount of curved modes increases.

An analysis of the coding efficiency and computation complexity indicates $\theta \leq 6$ as a good trade-off point. Regardless, it is essential to notice that we have tested all modes with RDO, and we have applied

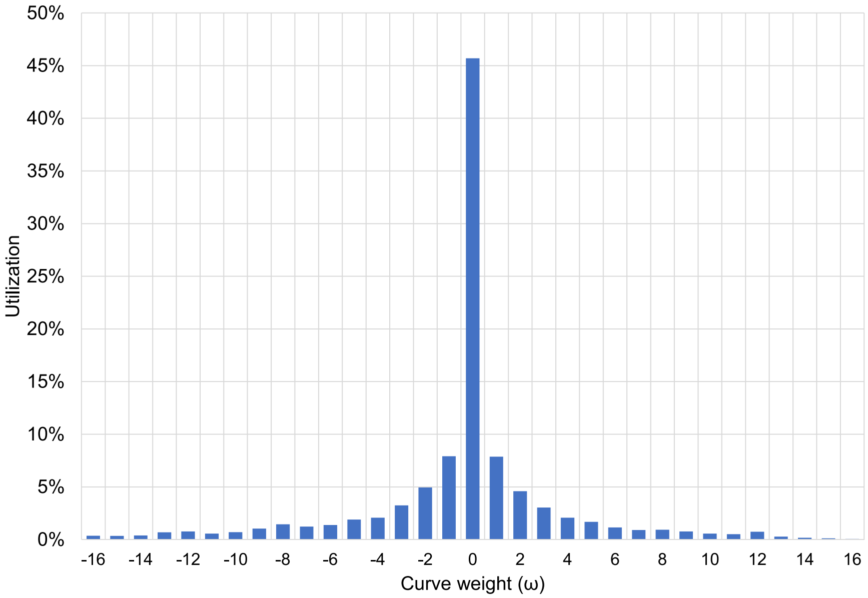


Fig. 6. Average utilization of curves regarding angular modes for all test sequences with $\theta = 32$. Low accentuated curves have higher utilization, indicating that high displacement values offer fewer benefits for coding efficiency.

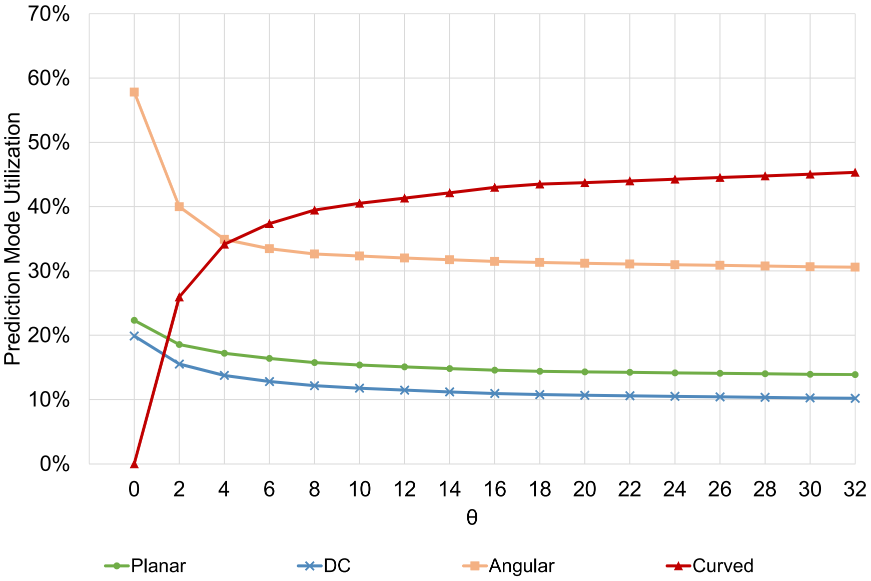


Fig. 7. Sample encoding when using the All-Intra-8 configuration. Even with an overabundance of curved modes for $\theta > 6$, a saturation point is noticeable as the utilization of curves stabilizes around 40%.

all curve displacement values to all block sizes, from 64 × 64 down to 4 × 4, meaning that we did not use any heuristic to reduce the number of evaluated curves.

Regarding the decoder complexity, our model does not result in additional computational effort. Once ω is captured from the bitstream, the PU reconstruction works similarly to the original encoder, with the difference of Eq. (14) and Eq. (15) for computing the sample values.

5.4. Comparison with state-of-the-art

We compare our results for $\theta \leq 8$ with state-of-the-art regarding improvements to HEVC intra-frame prediction since our method was evaluated using this standard as the baseline. We analyzed the works of [31], which employed HM 16.9, for MRL and its Fast variant (MRL Fast), along with Chen's improved intra-prediction encoding [28], which employed HM 12.0. The difference in terms of HM software versions does not allow a totally precise comparison, but this comparison is still possible, since all works are using the CTCs and all HM versions implement the same intra-prediction tools. Table 5 shows the experimental results of our work and the state-of-the-art.

Although we would like to compare our model for both coding configurations, the related works only provided results regarding All-Intra-8, while the Random-Access results are absent.

Our method reached similar coding efficiency gains to those found in related works when considering the same computational costs. However, our method can reach the highest encoding efficiency than all related works when using a higher computational cost. Another important issue here is that the presented methods are not mutually exclusive, and then, they can be grouped to reach even higher encoding efficiency.

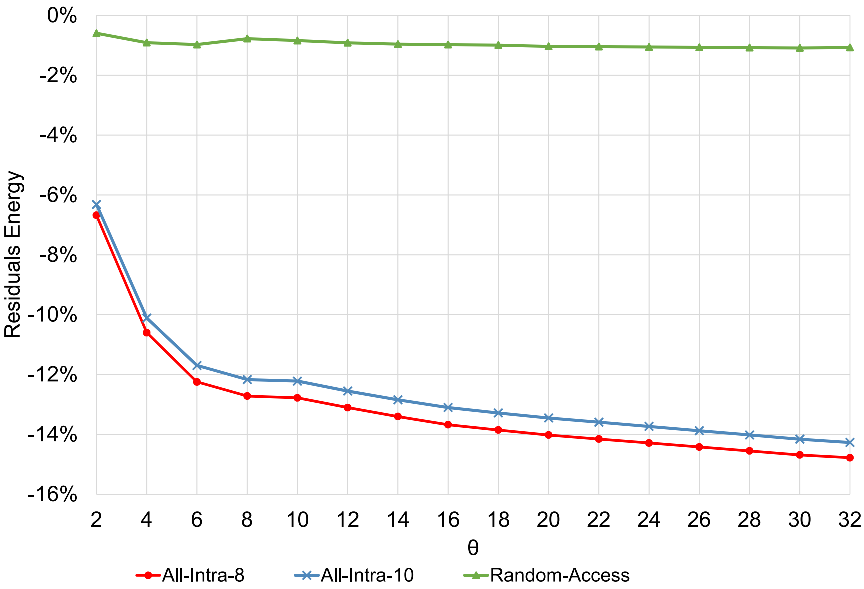
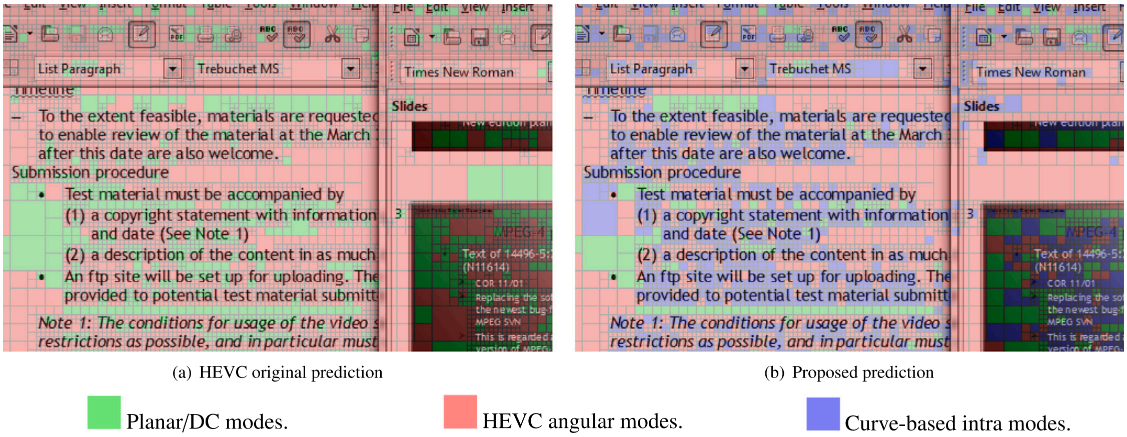


Fig. 8. Average residual energy reduction for all test configurations.



– To the extent feasible, materials are requested to enable review of the material at the March after this date are also welcome.

Submission procedure

- Test material must be accompanied by
(1) a copyright statement with information and date (See Note 1)
(2) a description of the content in as much
- An ftp site will be set up for uploading. The provided to potential test material submitt

Note 1: The conditions for usage of the video s restrictions as possible, and in particular must

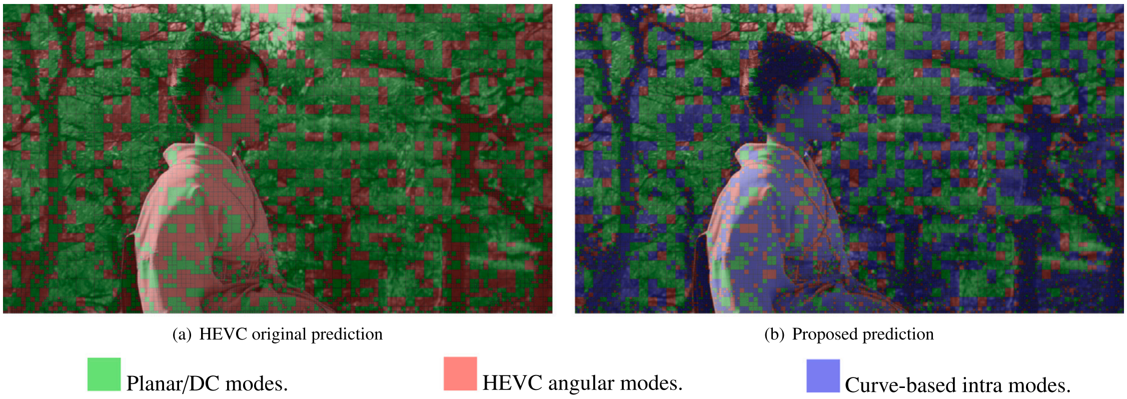
Text of 14496-5: (N11614)
COR 11/01
Replacing the so the newest bug-f MPEG SVN
This is regarded version of MPEG-

(a) HEVC original prediction

(b) Proposed prediction

🟩 Planar/DC modes. 🟥 HEVC angular modes. 🟦 Curve-based intra modes.

Fig. 9. A slice of one frame from the test sequence *SlideEditing*. The original HEVC prediction is shown in (a), while the proposed curved prediction is shown in (b). Green shaded blocks correspond to Planar and DC encoded PUs and red predicted blocks to angular predicted PUs; meanwhile, the blue shaded blocks correspond to curved modes, which concentrate around the text content of the video frame.



(a) HEVC original prediction

(b) Proposed prediction

🟩 Planar/DC modes. 🟥 HEVC angular modes. 🟦 Curve-based intra modes.

Fig. 10. A frame from the test sequence *Kimono*. The original HEVC prediction is shown in (a), while the proposed curved prediction is shown in (b). Green shaded blocks correspond to Planar and DC encoded PUs and red predicted blocks to angular predicted PUs; meanwhile, the blue shaded blocks correspond to curved modes, concentrating around curved textures like hair, tree branches, and clothing shadows.

Table 4
Summary of BD-Rate results for the All-Intra and Random-Access configurations and θ values.

All-Intra-8

Class	*θ*																**Average**
	2	**4**	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**	**22**	**24**	**26**	**28**	**30**	**32**	
Class A1	−1.36%	−2.05%	−2.45%	−2.63%	−2.65%	−2.78%	−2.88%	−2.96%	−3.05%	−3.12%	−3.16%	−3.21%	−3.25%	−3.29%	−3.34%	−3.36%	−2.85%
Class A2	−1.50%	−2.20%	−2.54%	−2.59%	−2.59%	−2.64%	−2.66%	−2.65%	−2.65%	−2.67%	−2.67%	−2.66%	−2.66%	−2.66%	−2.66%	−2.63%	−2.54%
Class B	−0.93%	−1.42%	−1.69%	−1.73%	−1.71%	−1.75%	−1.77%	−1.76%	−1.79%	−1.81%	−1.81%	−1.80%	−1.80%	−1.80%	−1.80%	−1.77%	−1.70%
Class C	−1.74%	−2.93%	−3.44%	−3.57%	−3.54%	−3.67%	−3.77%	−3.85%	−3.91%	−3.98%	−4.04%	−4.09%	−4.14%	−4.19%	−4.24%	−4.27%	−3.71%
Class D	−1.88%	−3.00%	−3.45%	−3.66%	−3.59%	−3.70%	−3.79%	−3.85%	−3.92%	−3.98%	−4.01%	−4.05%	−4.09%	−4.12%	−4.16%	−4.17%	−3.71%
Class E	−1.21%	−1.72%	−1.97%	−1.89%	−1.84%	−1.82%	−1.77%	−1.69%	−1.62%	−1.62%	−1.59%	−1.56%	−1.53%	−1.51%	−1.48%	−1.43%	−1.64%
Class F	−1.25%	−2.31%	−2.75%	−2.77%	−2.73%	−2.85%	−2.95%	−3.02%	−3.05%	−3.11%	−3.15%	−3.19%	−3.23%	−3.27%	−3.31%	−3.33%	−2.89%
Overall	−1.40%	−2.23%	−2.61%	−2.69%	−2.66%	−2.74%	−2.80%	−2.83%	−2.86%	−2.90%	−2.93%	−2.95%	−2.97%	−2.99%	−3.01%	−3.01%	–

All-Intra-10

Class	*θ*																**Average**
	2	**4**	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**	**22**	**24**	**26**	**28**	**30**	**32**	
Class A1	−1.13%	−1.70%	−2.04%	−2.17%	−2.18%	−2.29%	−2.37%	−2.42%	−2.49%	−2.56%	−2.60%	−2.65%	−2.69%	−2.74%	−2.79%	−2.81%	−2.35%
Class A2	−1.32%	−1.96%	−2.26%	−2.28%	−2.27%	−2.31%	−2.32%	−2.30%	−2.30%	−2.32%	−2.31%	−2.30%	−2.30%	−2.29%	−2.29%	−2.26%	−2.21%
Class B	−0.85%	−1.29%	−1.53%	−1.53%	−1.51%	−1.55%	−1.56%	−1.54%	−1.53%	−1.55%	−1.55%	−1.55%	−1.55%	−1.54%	−1.54%	−1.52%	−1.48%
Class C	−1.68%	−2.84%	−3.34%	−3.45%	−3.43%	−3.55%	−3.65%	−3.71%	−3.79%	−3.86%	−3.91%	−3.96%	−4.01%	−4.05%	−4.10%	−4.13%	−3.59%
Class D	−1.85%	−2.97%	−3.41%	−3.58%	−3.54%	−3.65%	−3.73%	−3.78%	−3.84%	−3.90%	−3.94%	−3.97%	−4.01%	−4.05%	−4.08%	−4.10%	−3.65%
Class E	−1.14%	−1.61%	−1.80%	−1.68%	−1.62%	−1.58%	−1.52%	−1.42%	−1.34%	−1.33%	−1.30%	−1.26%	−1.23%	−1.20%	−1.16%	−1.11%	−1.39%
Class F	−1.17%	−2.23%	−2.64%	−2.67%	−2.61%	−2.73%	−2.81%	−2.87%	−2.91%	−2.96%	−3.00%	−3.03%	−3.06%	−3.10%	−3.13%	−3.14%	−2.75%
Overall	−1.30%	−2.09%	−2.44%	−2.49%	−2.47%	−2.54%	−2.59%	−2.60%	−2.63%	−2.67%	−2.69%	−2.71%	−2.73%	−2.74%	−2.76%	−2.76%	–

Random-Access

Class	*θ*																**Average**
	2	**4**	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**	**22**	**24**	**26**	**28**	**30**	**32**	
Class A1	−0.10%	−0.09%	−0.02%	0.08%	0.11%	0.13%	0.19%	0.27%	0.32%	0.32%	0.36%	0.39%	0.42%	0.46%	0.49%	0.55%	0.24%
Class A2	−0.07%	0.07%	0.15%	0.29%	0.33%	0.39%	0.48%	0.61%	0.73%	0.76%	0.82%	0.88%	0.94%	1.00%	1.06%	1.14%	0.60%
Class B	−0.26%	−0.34%	−0.37%	−0.33%	−0.31%	−0.31%	−0.28%	−0.23%	−0.20%	−0.20%	−0.18%	−0.16%	−0.13%	−0.11%	−0.09%	−0.04%	−0.22%
Class C	−0.76%	−1.26%	−1.47%	−1.49%	−1.49%	−1.54%	−1.55%	−1.54%	−1.57%	−1.61%	−1.62%	−1.62%	−1.63%	−1.64%	−1.65%	−1.64%	−1.51%
Class D	−0.81%	−1.31%	−1.51%	−1.55%	−1.52%	−1.57%	−1.60%	−1.59%	−1.58%	−1.62%	−1.63%	−1.64%	−1.66%	−1.67%	−1.68%	−1.67%	−1.54%
Class F	−0.81%	−1.46%	−1.72%	−1.82%	−1.81%	−1.88%	−1.93%	−1.94%	−1.96%	−2.02%	−2.05%	−2.08%	−2.11%	−2.14%	−2.17%	−2.17%	−1.88%
Overall	−0.49%	−0.78%	−0.88%	−0.87%	−0.84%	−0.86%	−0.86%	−0.82%	−0.79%	−0.81%	−0.81%	−0.80%	−0.79%	−0.78%	−0.77%	−0.74%	–

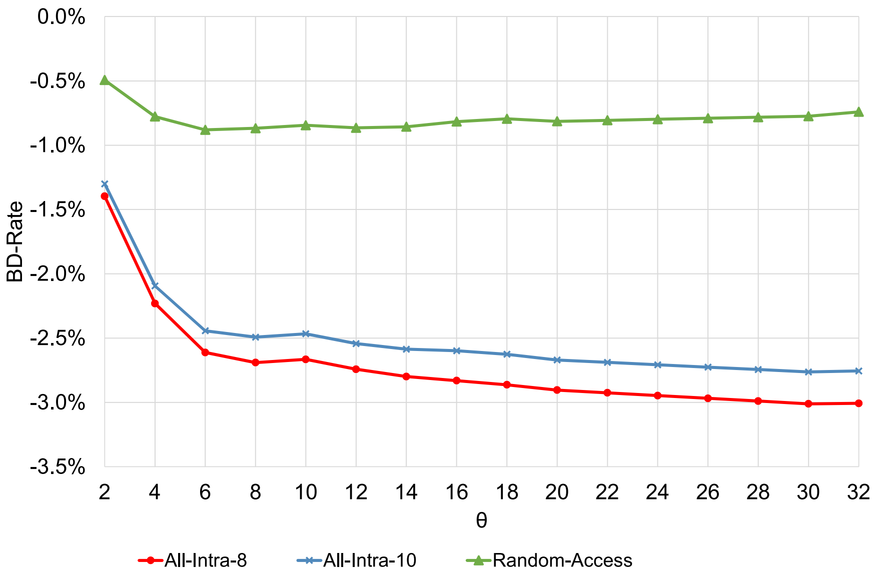


Fig. 11. BD-Rate reduction with θ ranging from 2 to 32, for each test sequence class.

6. Conclusion and future work

This article presented a novel sample displacement model for intra-frame prediction in current and future video coding standards. Our model enables a PU to represent curved textures instead of typical straight lines, offering better prediction quality with lowered residual energy.

We implemented our proposal inside the HEVC reference software as a new set of intra-frame prediction modes to be applied together with the default angular modes. The experiments unveiled that the new curved modes increased the encoding efficiency, especially when All-Intra-8 and All-Intra-10 configurations are considered. The use of a low amount of curve displacements offers a good trade-off in BD-Rate gains versus the encoding complexity required by our method, with $\theta \leq 6$ offering the most promising results. Additional displacement

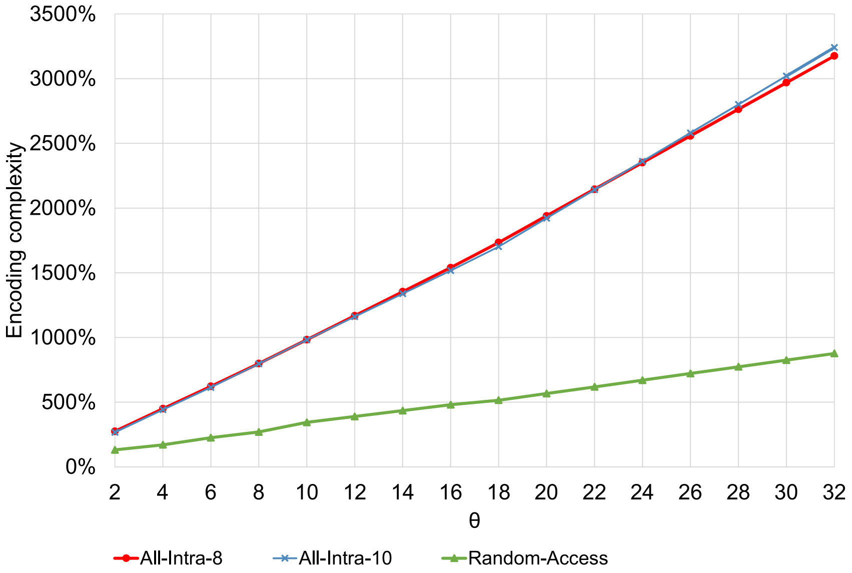


Fig. 12. Average encoding complexity penalty for all encoding configurations.

Table 5
Comparison with state-of-the-art regarding BD-Rate reduction for the All-Intra-8 configuration.

Sequence	State-of-the-art			Proposed work			
	MRL	MRL (Fast)	Chen 2017	$\theta = 2$	$\theta = 4$	$\theta = 6$	$\theta = 8$
Class A1	Class A: −2.18%	Class A: −1.85%	Class A: −1.84%	−1.36%	−2.05%	−2.45%	−2.63%
Class A2				−1.50%	−2.20%	−2.54%	−2.59%
Class B	−2.30%	−1.92%	−1.85%	−0.93%	−1.42%	−1.69%	−1.73%
Class C	−2.48%	−2.05%	−1.94%	−1.74%	−2.93%	−3.44%	−3.57%
Class D	−2.03%	−1.68%	−1.70%	−1.88%	−3.00%	−3.45%	−3.66%
Class E	−2.17%	−1.90%	−2.17%	−1.21%	−1.72%	−1.97%	−1.89%
Class F	–	–	−1.94%	−1.25%	−2.31%	−2.75%	−2.77%
Overall	−2.40%	−2.00%	−1.89%	−1.40%	−2.23%	−2.61%	−2.69%
Encoding time	462.00%	214.00%	230.00%	275.30%	449.45%	622.50%	798.57%

values have shown increased coding efficiency, albeit at much higher computational costs.

In future work, we aim to optimize the number of available displacement values according to the PU size, as we currently perform an exhaustive evaluation of possibilities for all PU sizes. Smaller PUs may not benefit from high displacement values, while larger PUs could benefit from additional displacement options. Limiting these extra curves to specific PU sizes may result in coding efficiency gains while lowering the computational overhead. We also intend to evaluate the proposed method with other standards like VVC and AV1. Since the curved modes were conceived to be additional intra-prediction modes, these modes can be evaluated together with new intra-frame prediction tools proposed by these new encoders.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nivel Superior — Brasil (CAPES) — Finance Code 001.

References

[1] A. Kalampogia, P. Koutsakis, H.264 and h.265 video bandwidth prediction, IEEE Trans. Multimedia 20 (1) (2018) 171–182.

[2] G. Sullivan, J. Ohm, W.-J. Han, T. Wiegand, Overview of the high efficiency video coding (HEVC) standard, IEEE Trans. Circuits Syst. Video Technol. 22 (12) (2012) 1649–1668.

[3] D. Mukherjee, J. Bankoski, A. Grange, J. Han, J. Koleszar, P. Wilkins, Y. Xu, R. Bultje, The latest open-source video codec VP9-an overview and preliminary results, in: Picture Coding Symposium, PCS, IEEE, 2013, pp. 390–393.

[4] A. for Open Media, AV1 bitstream & decoding process specification, 2019, https://aomediacodec.github.io/av1-spec/av1-spec.pdf. (Accessed 19 March 2020).

[5] B. Bross, J. Chen, S. Liu, Versatile Video Coding (Draft 5), Technical Report, JVET-N1001-V10, The Moving Picture Experts Group, 2019.

[6] I.-K. Kim, J. Min, T. Lee, W.-J. Han, J. Park, Block partitioning structure in the HEVC standard, IEEE Trans. Circuits Syst. Video Technol. 22 (12) (2012) 1697–1706.

[7] D. Marpe, H. Schwarz, T. Wiegand, Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard, IEEE Trans. Circuits Syst. Video Technol. 13 (7) (2003) 620–636.

[8] T. Wiegand, H. Schwarz, Video coding: part II of fundamentals of source and video coding, Found. Trends in Signal Process. 10 (1–3) (2016) 1–346.

[9] J. Lainema, F. Bossen, W.-J. Han, J. Min, K. Ugur, Intra coding of the HEVC standard, IEEE Trans. Circuits Syst. Video Technol. 22 (12) (2012) 1792–1801.

[10] C. Rhee, K. Lee, T. Kim, H.-J. Lee, A survey of fast mode decision algorithms for inter-prediction and their applications to high efficiency video coding, IEEE Trans. Consum. Electron. 58 (4) (2012) 1375–1383.

[11] M. Winken, P. Helle, D. Marpe, H. Schwarz, T. Wiegand, Transform coding in the HEVC test model, in: International Conference on Image Processing, ICIP, IEEE, 2011, pp. 3693–3696.

[12] R. Fernandes, G. Sanchez, R. Cataldo, T. Webber, C. Marcon, Efficient HEVC intra-frame prediction using curved angular modes, IET Electron. Lett. 54 (21) (2018) 1214–1216.

[13] G. Tech, Y. Chen, K. Müller, J. Ohm, A. Vetro, Y. Wang, Overview of the multiview and 3D extensions of high efficiency video coding, IEEE Trans. Circuits Syst. Video Technol. 26 (1) (2016) 35–49.

[14] L. Zhao, K. Zhou, J. Guo, S. Wang, T. Lin, A universal string matching approach to screen content coding, IEEE Trans. Multimed. 20 (4) (2018) 796–809.

[15] E. Alshina, G. Sullivan, J.-R. Ohm, J. Boyce, J. Chen, JVET-D1001: Algorithm Description of Joint Exploration Test Model 4, Technical Report, Joint Video Exploration Team, 2018.

[16] Y. Chang, H. Jhu, H. Jian, L. Zhao, X. Zhao, et al., Intra prediction using multiple reference lines for the versatile video coding standard, in: Applications of Digital Image Processing, International Society for Optics and Photonics, 2019, pp. 302–309.

[17] A. Said, X. Zhao, M. Karczewicz, J. Chen, F. Zou, Position dependent prediction combination for intra-frame video coding, in: International Conference on Image Processing, ICIP, IEEE, 2016, pp. 534–538.

[18] L. Zhao, X. Zhao, S. Liu, X. Li, J. Lainema, G. Rath, F. Urban, F. Racapé, Wide angular intra prediction for versatile video coding, in: Data Compression Conference, DCC, IEEE, 2019, pp. 53–62.

[19] L. Shen, Z. Zhang, P. An, Fast CU size decision and mode decision algorithm for HEVC intra coding, IEEE Trans. Consum. Electron. 59 (1) (2013) 207–213.

[20] X. Liu, Y. Liu, P. Wang, C. Lai, H. Chao, An adaptive mode decision algorithm based on video texture characteristics for HEVC intra prediction, IEEE Trans. Circuits Syst. Video Technol. 27 (8) (2017) 1737–1748.

[21] L. Feng, M. Dai, C.-L. Zhao, J.-Y. Xiong, Fast prediction unit selection method for HEVC intra prediction based on salient regions, Optoelectron. Lett. 12 (4) (2016) 316–320.

[22] H. Azgin, E. Kalali, I. Hamzaoglu, A computation and energy reduction technique for HEVC intra prediction, IEEE Trans. Consum. Electron. 63 (1) (2017) 36–43.

[23] H. Yuan, Y. Chang, Z. Lu, M. Li, Fast and efficient intraprediction method for H. 264/AVC, Opt. Eng. 49 (4) (2010) 3.

[24] H. Yuan, C. Guo, J. Liu, X. Wang, S. Kwong, Motion-homogeneous-based fast transcoding method from H.264/AVC to HEVC, IEEE Trans. Multimed. 19 (7) (2017) 1416–1430.

[25] K. Sharman, Common test conditions, 2018, https://jvet.hhi.fraunhofer.de/. (Accessed 19 March 2020).

[26] L. Lucas, N. Rodrigues, C. Pagliari, E. da Silva, S. de Faria, Sparse least-squares prediction for intra image coding, in: International Conference on Image Processing, ICIP, IEEE, 2015, pp. 1115–1119.

[27] S. Matsuo, S. Takamura, A. Shimizu, Modification of intra angular prediction in HEVC, in: Asia Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA ASC, IEEE, 2012, pp. 1–4.

[28] C. Chen, S. Zhu, B. Zeng, M. Gabbouj, A new block-based method for HEVC intra coding, IEEE Trans. Circuits Syst. Video Technol. 27 (8) (2017) 1727–1736.

[29] S. Zhu, B. Zeng, Constrained quantization in the transform domain with applications in arbitrarily-shaped object coding, IEEE Trans. Circuits Syst. Video Technol. 20 (11) (2010) 1385–1394.

[30] G. Bjøntegaard, Calculation of average PSNR differences between RD-curves, VCEG-M33 (2001).

[31] J. Li, B. Li, J. Xu, R. Xiong, Intra prediction using multiple reference lines for video coding, in: Data Compression Conference, DCC, IEEE, 2017, pp. 221–230.

[32] G. Sullivan, T. Wiegand, Rate-distortion optimization for video compression, IEEE Signal Process. Mag. 15 (6) (1998) 74–90.

[33] J. Lainema, K. Ugur, Angular intra prediction in high efficiency video coding (HEVC), in: International Workshop on Multimedia Signal Processing, MMSP, IEEE, 2011, pp. 1–5.

[34] V. Sze, M. Budagavi, G. Sullivan, High Efficiency Video Coding (HEVC): Algorithms and Architectures, Springer Publishing Company, Incorporated, 2014.

[35] L. Zhao, L. Zhang, S. Ma, D. Zhao, Fast mode decision algorithm for intra prediction in HEVC, in: Visual Communications and Image Processing, VCIP, IEEE, 2011, pp. 1–4.

[36] H. Sun, Z. Cheng, M. Takeuchi, J. Katto, Enhanced intra prediction for video coding by using multiple neural networks, IEEE Trans. Multimedia 1 (1) (2020) 16.

[37] T.J.C.T. on Video Coding (JCT-VC), HEVC test model HM-16.8. ITU-t VCEG and ISO/IEC MPEG (JCT-VC), 2018, https://hevc.hhi.fraunhofer.de/svn/svnHEVCSoftware/. (Accessed 19 March 2020).