

Article

Fusion-Based Versatile Video Coding Intra Prediction Algorithm with Template Matching and Linear Prediction

Dan Luo ¹, Shuhua Xiong ^{1,*}, Chao Ren ¹ , Raymond Edward Sheriff ²  and Xiaohai He ¹

¹ College of Electronics and Information Engineering, Sichuan University, No. 24 South Section 1, Yihuan Road, Chengdu 610065, China

² Department of Computer Science, Edge Hill University, Ormskirk L39 4QP, UK

* Correspondence: xiongsh@scu.edu.cn

Abstract: The new generation video coding standard Versatile Video Coding (VVC) has adopted many novel technologies to improve compression performance, and consequently, remarkable results have been achieved. In practical applications, less data, in terms of bitrate, would reduce the burden of the sensors and improve their performance. Hence, to further enhance the intra compression performance of VVC, we propose a fusion-based intra prediction algorithm in this paper. Specifically, to better predict areas with similar texture information, we propose a fusion-based adaptive template matching method, which directly takes the error between reference and objective templates into account. Furthermore, to better utilize the correlation between reference pixels and the pixels to be predicted, we propose a fusion-based linear prediction method, which can compensate for the deficiency of single linear prediction. We implemented our algorithm on top of the VVC Test Model (VTM) 9.1. When compared with the VVC, our proposed fusion-based algorithm saves a bitrate of 0.89%, 0.84%, and 0.90% on average for the Y, Cb, and Cr components, respectively. In addition, when compared with some other existing works, our algorithm showed superior performance in bitrate savings.

Keywords: Versatile Video Coding; intra prediction; template matching; linear prediction



Citation: 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. <https://doi.org/10.3390/s22165977>

Academic Editors: Byung-Gyu Kim and Dongsan Jun

Received: 25 June 2022

Accepted: 5 August 2022

Published: 10 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 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 (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, Internet traffic comprising multimedia data has increased rapidly, with 80% expected to be of video content in the coming years. Therefore, efficient video compression is very important. The new generation video coding standard Versatile Video Coding (VVC) [1] was officially released in July 2020. On the basis of the High Efficient Video Coding (HEVC) [2] standard, VVC has introduced many advanced technologies to further improve compression performance. Simultaneously, VVC supports a wide variety of video types such as High Definition (HD) and Ultra HD (UHD) resolution, Wide Color Gamut (WCG) video sequences, Virtual Reality (VR) videos, ultra-low delay applications, and so on.

Currently, VVC continues to employ a block-based hybrid coding framework, while most of its coding tools have been improved. In terms of block division, in addition to the traditional Quad Tree (QT), to make the partitioning results more suitable for video content, Binary Tree (BT) and Ternary Tree (TT) [3] have also been introduced. For intra prediction, the angle prediction modes are more detailed, and Position Dependent Intra Prediction Combination (PDPC) [4,5], Multiple Reference Line (MRL) [6,7], Intra Sub-Partition (ISP) [8,9], Cross Component Linear Model (CCLM) [10–12], Matrix Weighted Intra Prediction (MIP) [13,14] and Wide-Angle Intra Prediction (WAIP) [15–17] have been utilized to optimize intra prediction performance. Bi-Directional Optical Flow (BDOF) [18,19], Decoder-side Motion Vector Refinement (DMVR) [20], and Affine Motion Compensation (AMC) [21,22] have been used to improve inter prediction precision. Multiple Transform

Selection (MTS) [23], Sub-Block Transform (SBT) [24], and Low Frequency Non-Separable Transform (LFNST) [25] have been employed to further eliminate frequency redundancy. In addition, many other modules have also been optimized.

Intra prediction predicts the current pixels by using the reconstructed pixels of the current frame to remove spatial redundancy. Recently, many works on intra prediction have been performed. Schneider et al. [26] proposed an algorithm named Sparse Coding-Based Intra Prediction (SCIP) to further improve the performance of VVC. To better predict the areas with complex textures, Li et al. [27] presented an improved intra prediction mode combination method and introduced an efficient mode coding method of syntax elements to enhance the coding performance. Yoon et al. [28] designed a method to obtain the parameters of CCLM more precisely, which compensated for the coding loss of the simplified CCLM mode. To fully utilize the advantages of Intra Block Copy (IBC) and palette coding, Zhu et al. [29] designed a compound palette mode to improve the performance of VVC on screen content coding. Since IBC cannot deal well with geometric transformations, Jayasingam et al. [30] extended IBC to adapt contents with zooms, rotations, and stretches. In [31], Weng et al. presented an L-shape-based iterative algorithm to improve intra prediction accuracy, and a residual median edge detection method was also proposed to address edge information. In [32], Yoon et al. exploited the number of occurrences of the modes in the neighboring blocks to extend the Most Probable Mode (MPM). Wang et al. [33] designed a Sample Adaptive Offset (SAO) acceleration method to reduce the complexity of VVC. Saha et al. [34] analyzed the decoder complexity of VVC on two different platforms.

Intra prediction plays an important role in video coding, and effectively compressing data can reduce the workload of the sensors. Although there are already some works that have achieved good results, there have been few that have taken the different features of different blocks and different modes into account. Hence, there is still room for further improvement. In this paper, we propose a fusion-based algorithm. The main contributions of our work are as follows:

- (1) We proposed a fusion-based adaptive template matching method. Its core idea is adaptive template matching. This method sufficiently considers the influence of matching errors on prediction results. The mode using this method is named mode 67;
- (2) We designed a fusion-based linear prediction method. Its core idea is linear prediction. This method fully considers the linear relationship between reference pixels and the pixels to be predicted, and the correlation between different models. The mode utilizing this method is named mode 68.

The remainder of the paper is organized as follows. Section 2 introduces the related works. In Section 3, our proposed fusion-based algorithm is presented in detail. In Section 4, we conduct some experiments to verify the effectiveness of our proposed algorithm, and conclusions are drawn in Section 5.

2. Related Works

The Template Matching Prediction (TMP) algorithm is a widely used method in video coding. The TMP algorithm takes the reference pixels of the current Prediction Unit (PU) block as a reference template. In the reconstructed area, a certain criterion is employed to select some candidate templates with the least error, from which the final prediction value is obtained. To reduce the effect of compression noise, Tan et al. [35] proposed using the average value of several candidate templates. Gayathri et al. [36] presented a region-based TMP algorithm that could reduce the complexity and obtain good performance. In [37], Gayathri et al. further decreased the memory requirements and number of computations at the decoder side. Considering that only the local reference samples could not deal well with complex areas, Lei et al. [38] designed a two-step progressive method to use both local (derived by the high frequency coefficients) and non-local information (obtained through TMP). These methods achieved good performance, however, simply averaging does not take the different importance of different blocks into account, and a linear approach is not

adaptive and does not directly consider the error. To better predict PUs with rich texture information, we propose a fusion-based adaptive template matching method in this paper.

Linear prediction assumes a linear relationship between reference samples and the samples to be predicted. The final prediction value is obtained by constructing a linear function. Typically, CCLM linearly obtains the chroma prediction value through the corresponding reconstructed luminance samples. The process of CCLM is shown in Figure 1. First, the co-located luminance block is down-sampled. Then, the reference pixels of the current chroma block and the down-sampled luminance block are used to calculate the parameters by constructing a linear function. Finally, the chroma prediction value is obtained by a linear function.

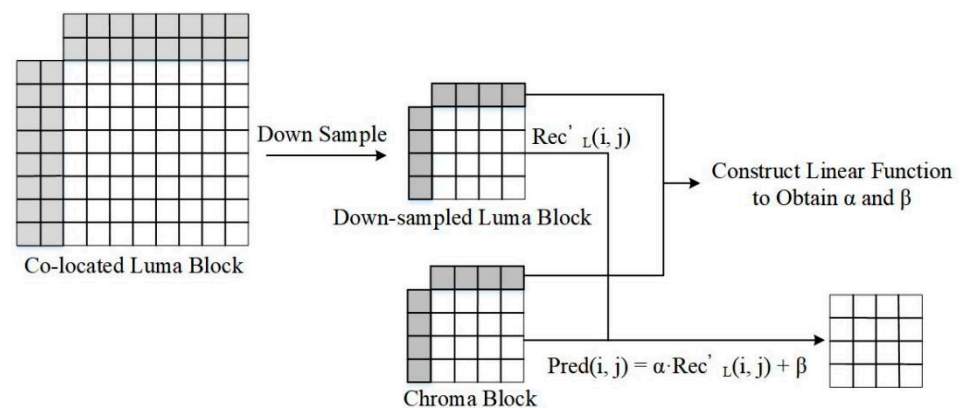


Figure 1. The process of CCLM in VVC.

As shown in Equation (1), $pred_c(i, j)$ is the chroma pixel to be predicted, and $rec'_L(i, j)$ denotes the down-sampled reconstructed luma pixel. (i, j) is the position and α , β are the parameters.

$$pred_c(i, j) = \alpha \times rec'_L(i, j) + \beta \quad (1)$$

Many related works have been undertaken based on CCLM. Ghaznavi-Youvalari et al. [39] merged CCLM with an angle mode derived from the corresponding luma block to improve the chroma prediction accuracy. Zhang et al. [40] introduced three methods including Multi-Model CCLM (MMCCLM), Multi-Filter CCLM (MFCCLM), and linear mode angle prediction to further enhance the coding efficiency of CCLM. However, there have been few linear prediction optimizations for the luminance component. In [41], Ghaznavi-Youvalari et al. presented a three-parameter linear function to improve the intra prediction performance. The parameters of this method were obtained based on a mean square Error (MSE) minimization approach from the reference pixels and their locations. This method achieves good efficiency, however, when the pixel to be predicted is far from the reference samples, the prediction performance decreases with the decline in the correlation. To address this problem, we proposed a fusion-based linear prediction method.

3. Proposed Method

As mentioned previously, our proposed fusion-based algorithm includes two parts: fusion-based adaptive template matching (mode 67) and fusion-based linear prediction (mode 68). The overall flowchart is depicted in Figure 2. We termed the VVC's original intra prediction modes (mode 0~mode 66) as the traditional/original modes. When conducting intra prediction, we have to first decide on whether the prediction mode is traditional. If the mode is traditional, the original mode of the VVC is utilized to obtain the prediction pixels. If the mode is not traditional, we then decide on whether the mode is 67 or 68. If it is mode 67, the fusion-based adaptive template matching is employed. If it is mode 68, fusion-based linear intra prediction is used. Some modes can be selected with less distortion, called

candidate modes, by rough calculation. Finally, the Rate–Distortion Cost (RDC) is utilized to determine the best mode from these candidate modes. Since the decoder can perform the same operations as the encoder to obtain the prediction and reconstructed values based on the mode number, we do not need to send extra flag bits.

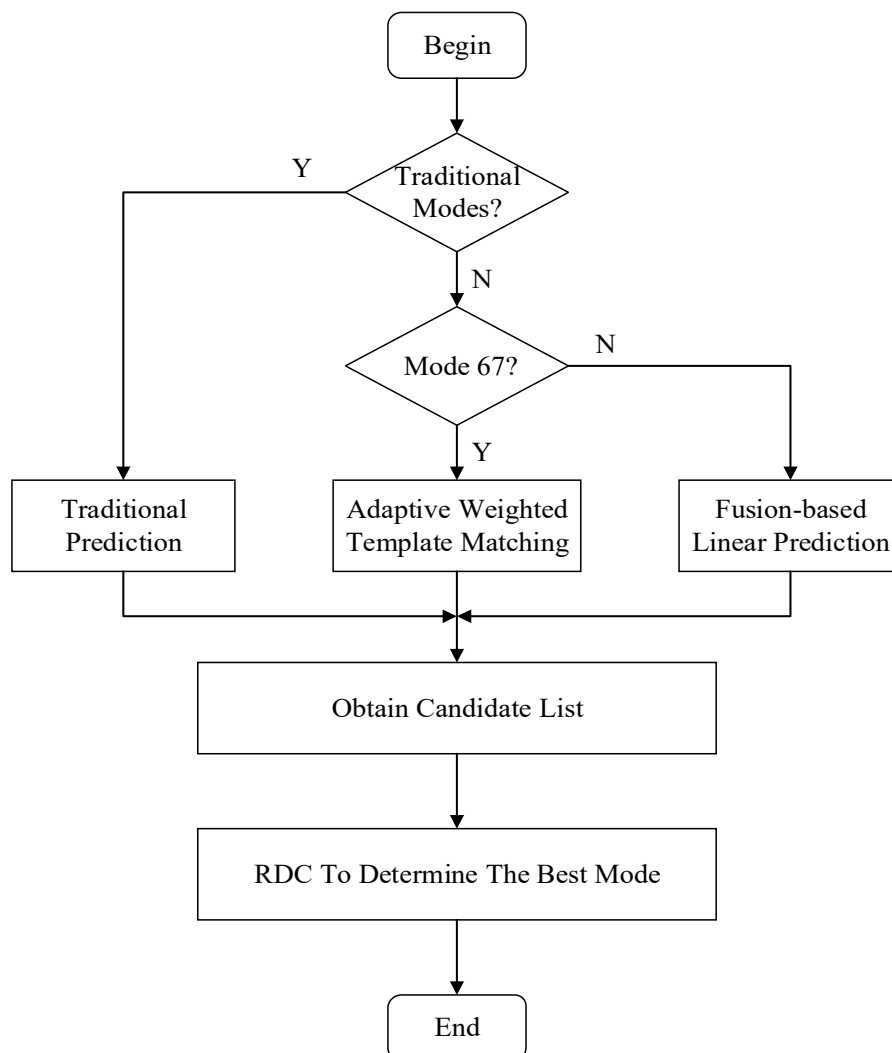


Figure 2. The overall framework of the proposed fusion-based algorithm.

3.1. Fusion-Based Adaptive Template Matching

Usually, high correlation and similar textures are prevalent with blocks, and the TMP algorithm can find candidate blocks with the least errors in the reconstructed area to better predict the region with similar textures. Since it searches and compares pixel by pixel, it usually performs well for blocks with similar textures. The process of template matching is shown in Figure 3, in which T_r is the reference template and T_i is the candidate template. B_p is the block to be predicted and B_i is the corresponding prediction block of T_i . w_i is the corresponding weighting factor of the candidate block. First, we find the candidate blocks T_i through T_r in the reconstructed area, and then B_i is utilized to obtain B_p .

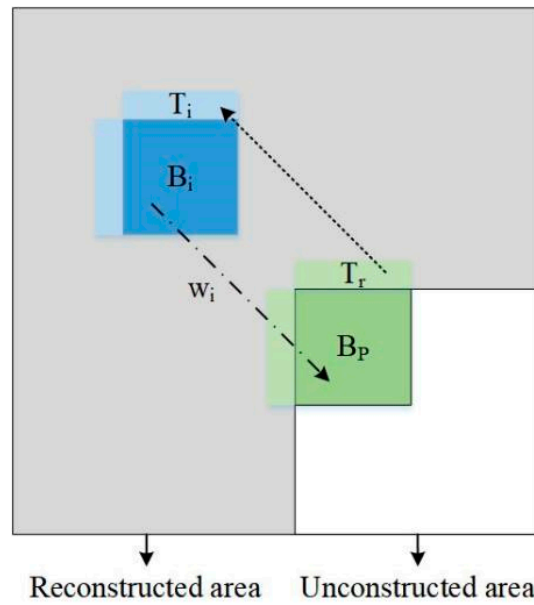


Figure 3. The process of template matching for block B_p .

Considering that the error between T_r and T_i has a large influence on the prediction result, we used it as a key basis for weight selection. To balance the compression performance and time complexity, we limited the searching area to 64×64 and chose the best four candidate templates through the MSE minimization criteria. The MSE is obtained by:

$$\sigma_i^2 = \frac{\sum_{j=0}^{N-1} (T_r(j) - T_i(j))^2}{N} \quad (2)$$

where σ_i^2 is the MSE between the reference template T_r and i -th candidate template T_i . N is the total number of pixels in T_r . If σ_i^2 is large, it implies that the difference between T_r and T_i is large, and the corresponding weight is set to be small. If σ_i^2 is small, it implies that the candidate template is close to the reference template, and the weight is set to be large. Consequently, the temporary weight can be obtained by introducing a logarithm function as follows:

$$\lambda_i = \ln(1 + e^{-\sigma_i^2}) \quad (3)$$

where λ_i is the temporary weight of the i -th candidate template. By normalization, the final weighting factor w_i of the i -th candidate template is obtained by:

$$w_i = \frac{\lambda_i}{\sum_{j=0}^3 \lambda_j} \quad (4)$$

Then, the final prediction value $p(x, y)$ can be obtained by:

$$p(x, y) = \sum_{i=0}^3 w_i \times B_i(x, y) \quad (5)$$

Generally, the TMP has a high time cost due to the pixel-by-pixel comparison and error calculation. It sacrifices time in exchange for performance gains. Hence, fusion-based adaptive template matching is only employed when the PU size is smaller or equal to 32×32 . Adding this limitation is important because when the PU size is large, the TMP algorithm will trade high time complexity for coding gain, which is not worthwhile.

Simultaneously, the texture information of large PUs is simple, so other modes can achieve good results.

3.2. Fusion-Based Linear Prediction

Linear prediction is a simple but efficient method because there is usually a high linear correlation between the pixels to be predicted and the reference samples. However, as the pixels to be predicted move away from the reference pixels, the correlation between them will weaken, and the performance of linear prediction will decline. To address this issue, we present a fusion-based linear prediction method based on Ghaznavi-Youvalari's work [41].

When the PU size is small, single three-parameter linear prediction can achieve good results. However, when the PU size is large, prediction accuracy declines with the weakening in correlation. Hence, we combined the three-parameter mode with planar mode to obtain the final prediction value. Planar mode obtains the prediction value by weighting the pixels in the horizontal and vertical directions, where the weights are related to the distance. The prediction process of the planar mode is shown in Figure 4.

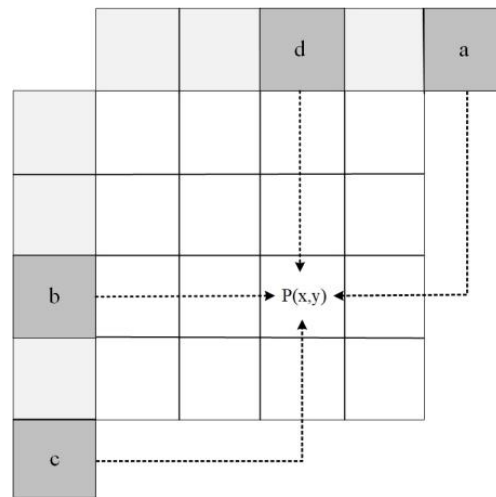


Figure 4. The prediction process of the planar mode based on reference pixels a , b , c , and d .

The prediction value of horizontal $p_h(x, y)$ is obtained by:

$$p_h(x, y) = (lw - x - 1) \times b + (x + 1) \times a \quad (6)$$

where (x, y) is the position of the current pixel. a , b are the reference samples and lw is the width of PU. Similarly, the prediction value of vertical $p_v(x, y)$ can be obtained by:

$$p_v(x, y) = (lh - y - 1) \times d + (y + 1) \times c \quad (7)$$

where c , d are the reference samples and lh is the height of PU.

Then, the final prediction value is the average of $p_h(x, y)$ and $p_v(x, y)$.

We can see that the prediction process of the planar mode is very close to linear prediction. Hence, the combination of planar mode and three-parameter linear mode can partially compensate for the shortcomings of a single linear prediction and improve the prediction precision.

When the PU size is smaller or equal to 32×32 , three-parameter linear prediction is used to obtain the final pixels. The linear function is constructed by:

$$p(x, y) = a_0 \times x + a_1 \times y + a_2 \quad (8)$$

where $p(x, y)$ is the prediction value. a_0 , a_1 , and a_2 are the parameters to be calculated. The specific solution is shown in [41].

When the PU size is larger than 32×32 , both three-parameter linear prediction and planar mode are utilized to obtain the final prediction value:

$$p(x, y) = w_1 \times p_{linear}(x, y) + w_2 \times p_{planar}(x, y) \quad (9)$$

where $p_{linear}(x, y)$ is the prediction value of three-parameter linear prediction and $p_{planar}(x, y)$ is that of the planar mode. w_1 and w_2 are the corresponding weighting factors. Since three-parameter linear prediction and planar mode adopt different prediction methods and they can provide different prediction information, they are equally important. Therefore, the two weighting factors were both set as 0.5.

4. Experimental Results and Analysis

4.1. Experimental Environment

To verify the effectiveness of our proposed fusion-based algorithm, we implemented it on top of VVC Test Model 9.1 (VTM9.1). Coding conditions were followed by the Joint Video Exploration Team (JVET) Common Test Condition (CTC) [42]. Sixteen test sequences with four kinds of resolution were utilized. The resolutions included 416×240 , 832×480 , 1280×720 , and 1920×1080 . We selected four common test QP values $\in \{22, 27, 32, 37\}$ to code video sequences in this work. Only the first 30 frames of each sequence were coded with All Intra (AI) configurations due to the experimental conditions.

4.2. Compression Performance

First, we tested the compression performance of our proposed fusion-based algorithm by comparing it with the VVC anchor. The Bjøntegaard Delta Rate (BD-Rate) method [43] was utilized to assess the compression performance. In addition to the respective BD-Rate of the three components, we also calculated the weighted BD-Rate of the three components by:

$$BDRate_{YUV} = \frac{(4 \times BDRate_Y + BDRate_U + BDRate_V)}{6} \quad (10)$$

where $BDRate_{YUV}$ is the weighted YUV BD-Rate. $BDRate_Y$, $BDRate_U$, and $BDRate_V$ represent the bitrate of Y, Cb, and Cr, respectively.

Considering that class F is the screen content sequence and the others are the natural content sequence, we made a distinction between them when calculating the average BD-Rate. If the BD-Rate is negative, it indicates that the performance of the VVC has improved. Otherwise, the coding performance of the VVC has deteriorated. As shown in Table 1, compared with the VVC anchor, our proposed fusion-based algorithm saved a bitrate of 0.89%, 0.84%, and 0.90% on average (up to 2.69%, 2.81%, and 2.81%) for components Y, Cb, and Cr, respectively. Simultaneously, our algorithm was particularly efficient for sequences such as “BasketballDrive” (1920×1080), “BQTerrace” (1920×1080), “Johnny” (1280×720), “BasketballPass” (416×240), and so on. This is mainly because there are many similar areas in these sequences. The correlation between blocks, and the correlation between reference pixels and the pixels to be predicted were relatively high. For most sequences in Class C and Class D, their texture content was very rich and the texture information varied greatly. For these two classes, the performance of our algorithm was not particularly outstanding since our proposed algorithm is suitable for videos with more similar texture areas. For almost all sequences, our proposed fusion-based algorithm achieved good results, which verified its effectiveness.

Table 1. The performance of our proposed algorithm compared with the VVC in BD-Rate.

Class	Sequence	Y/%	U/%	V/%	YUV/%
B (1920 × 1080)	BasketballDrive	−1.55	−1.46	−1.62	−1.55
	BQTerrace	−1.08	−1.15	−1.06	−1.08
	Cactus	−0.79	−0.62	−0.95	−0.79
	Average	−1.14	−1.08	−1.21	−1.14
C (832 × 480)	BQMall	−0.25	−0.20	−0.19	−0.23
	BasketballDrill	−0.41	−0.23	−0.64	−0.42
	PartyScene	−0.14	−0.12	−0.35	−0.17
	RaceHorses	0.02	−0.03	−0.31	−0.04
	Average	−0.20	−0.15	−0.37	−0.22
D (416 × 240)	BasketballPass	−0.53	−1.31	−1.50	−0.82
	RaceHorses	−0.01	−0.18	−0.32	−0.09
	BlowingBubbles	0.00	0.17	0.15	0.05
	BQSquare	−0.26	−0.23	−0.47	−0.29
	Average	−0.20	−0.39	−0.54	−0.29
E (1280 × 720)	Johnny	−2.69	−2.81	−2.81	−2.73
	FourPeople	−0.47	−0.55	−0.46	−0.48
	KristenAndSara	−0.81	−0.53	−0.10	−0.65
	Average	−1.32	−1.30	−1.12	−1.29
F (1280 × 720)	SlideShow	−0.72	−1.30	−0.53	−0.79
	SlideEditing	−2.28	−2.32	−2.32	−2.29
	BasketballDrillText (832 × 480)	−0.52	−0.40	−0.47	−0.49
	Average	−1.17	−1.34	−1.11	−1.19
Overall Average (Excluding D And F)		−0.89	−0.84	−0.90	−0.88
Overall Average (Including D And F)		−0.81	−0.85	−0.87	−0.82

Figures 5–7 illustrate the Rate–Distortion (RD) curves of some sequences in QP 22, 27, 32, 37. The horizontal axis is the bitrate and the vertical axis is the YUV-PSNR. If the curve of our proposed algorithm is above the VVC, this shows that the peak signal-to-noise ratio (PSNR) of our proposed algorithm is higher than that of VVC for the same bitrate, indicating that our proposed algorithm enhances the compression performance. Otherwise, our proposed algorithm deteriorated the performance. In Figure 6, the blue curve is VTM9.1, and the red one is proposed. We enlarged a small part (the position of the green rectangle) of each curve to show the comparison of results more intuitively. Clearly, our RD curves were higher than that of VVC. For areas with a similar texture, our proposed fusion-based algorithm could better model the relationship between the reference pixels and pixels to be predicted. Hence, for the same PSNR, our algorithm needs less bitrate, which means that our proposed algorithm improves the compression performance of VVC.

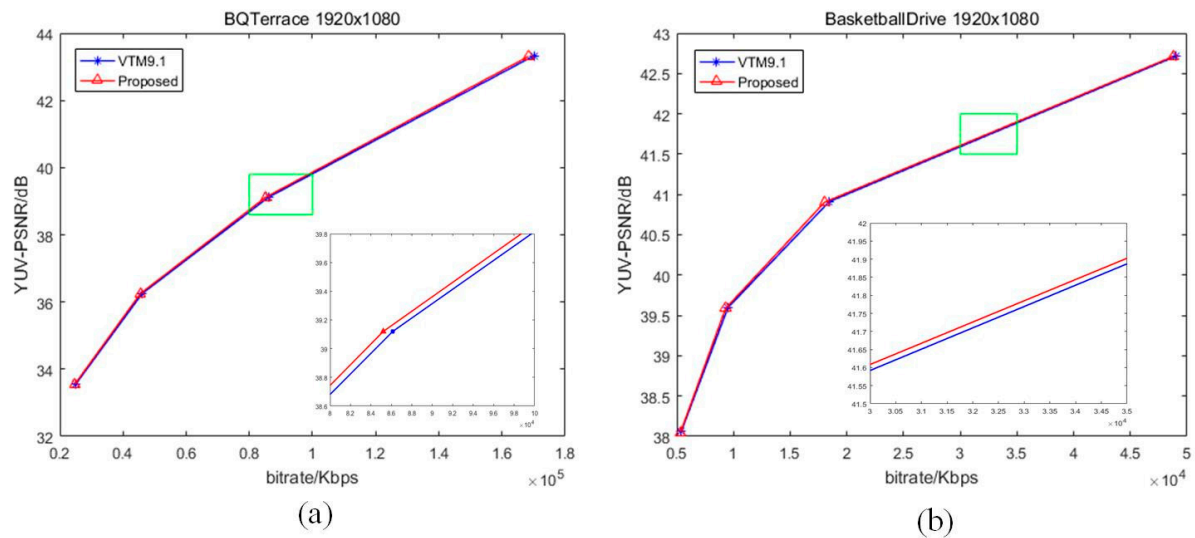


Figure 5. The RD curves of some sequences: (a) “BQTerrace” (1920×1080); (b) “BasketballDrive” (1920×1080).

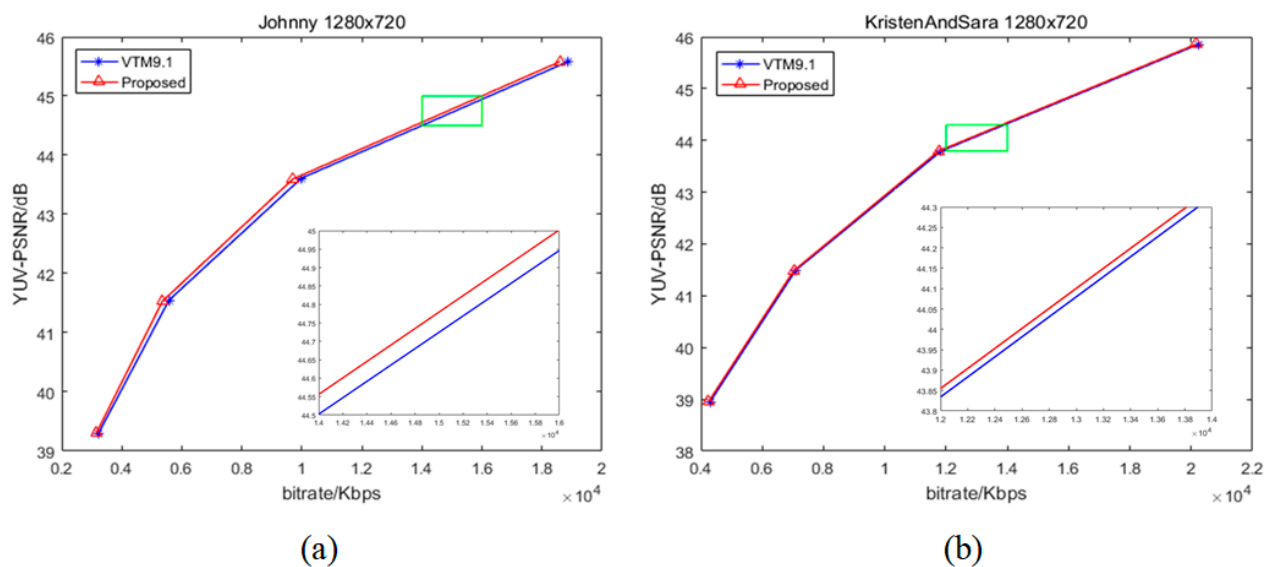


Figure 6. The RD curves of some sequences: (a) “Johnny” (1280×720); (b) “KristenAndSara” (1280×720).

Moreover, we compared our proposed fusion-based algorithm with some existing works to further verify the effectiveness of our work. This is detailed in Table 2. In most cases, the performance of our proposed algorithm was superior to the others. In [28], Yoon et al. optimized the CCLM algorithm and achieved good results, although their work only enhanced the channel correlation between the luminance and chroma components and did not consider the correlation of the stronger chroma components themselves. Therefore, overall, our algorithm performed better. In [32], Yoon et al. obtained the mode of the current PU block through the neighboring blocks and achieved good results. However, they only considered the correlation between the current block and its neighbors, not the correlation between the more distant blocks and the correlation between the reference pixels and the pixels to be predicted. Therefore, in general, our proposed algorithm has superior performance. In [26], Schneider et al. obtained the final prediction value through sparse coding and achieved good results, although the dictionary is finite, which limits the performance of their algorithm. Our

proposed fusion-based algorithm calculates the corresponding prediction value according to different PUs, which results in improved performance.

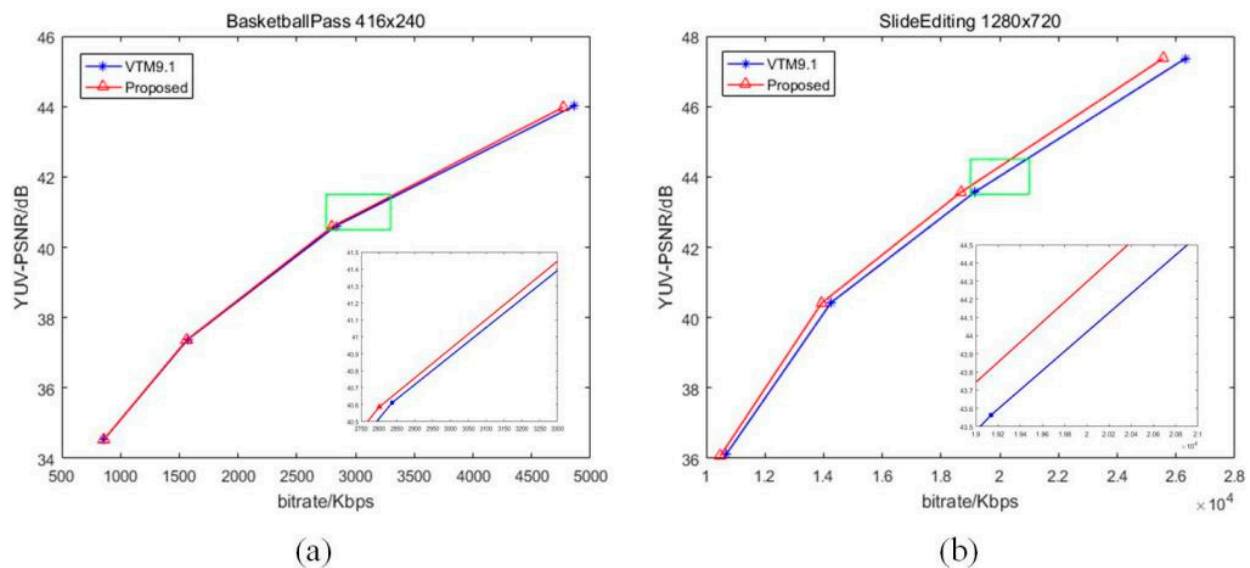


Figure 7. The RD curves of some sequences: (a) “BasketballPass” (416 × 240); (b) “SlideEditing” (1280 × 720).

Table 2. A comparison with other works in BD-Rate.

Class	BD-Rate/%	Yoon et al. [28]	Yoon et al. [32] (Test 3)	Schneider et al. [26]	Ours
B	Y	−0.04	−0.45	−0.33	−1.14
	U	−0.74	−0.39	−	−1.08
	V	−0.83	−0.38	−	−1.21
C	Y	−0.07	−0.47	−0.2	−0.20
	U	−0.72	−0.45	−	−0.15
	V	−0.85	−0.29	−	−0.37
E	Y	−0.02	−0.58	−0.3	−1.32
	U	−0.29	−0.45	−	−1.30
	V	−0.10	−0.52	−	−1.12
Average	Y	−0.04	−0.50	−0.28	−0.89
	U	−0.58	−0.43	−	−0.84
	V	−0.59	−0.40	−	−0.90

4.3. Mode Using Probability

Since our new added modes were in competition with the original VVC intra prediction modes, we calculated the using probability of our new modes, as shown in Equation (11). Five sequences were used to conduct this experiment.

$$\eta = \frac{N_{\text{mode}}}{N_{\text{total}}} \times 100\% \quad (11)$$

where η is the using probability. N_{mode} and N_{total} represent the using numbers of a single mode and all modes, respectively. If η is large, it shows that compared with other modes, our modes are chosen more often, indicating that our algorithm shows superior RD performance.

The using probability of mode 67 (the mode using fusion-based adaptive template matching) and mode 68 (the mode utilizing fusion-based linear prediction) in the luminance component are illustrated in Tables 3 and 4, respectively.

Table 3. The using probability of mode 67 in the luminance component.

Sequence	QP = 22	QP = 27	QP = 32	QP = 37	Average
BasketballPass	9.54%	7.48%	5.25%	4.06%	6.58%
BQMall	1.62%	1.48%	1.54%	1.44%	1.52%
Johnny	10.14%	11.03%	10.99%	9.84%	10.50%
BasketballDrive	5.24%	7.01%	9.16%	9.37%	7.70%
BQTerrace	8.27%	7.67%	7.20%	6.22%	7.34%

Table 4. The using probability of mode 68 in the luminance component.

Sequence	QP = 22	QP = 27	QP = 32	QP = 37	Average
BasketballPass	0.00%	0.00%	0.29%	0.28%	0.14%
BQMall	0.80%	0.10%	0.12%	0.08%	0.28%
Johnny	0.12%	0.13%	0.06%	0.00%	0.08%
BasketballDrive	0.08%	0.03%	0.06%	0.04%	0.05%
BQTerrace	0.22%	0.08%	0.08%	0.03%	0.10%

Both mode 67 and mode 68 were selected to some extent. Specifically, mode 67 was often chosen because of its good performance. Its maximum using probability reached 10.50% for “Johnny” (1280×720). The performance of mode 68 was not as appealing as mode 67. Since the TMP algorithm exploits the correlation between blocks and mode 68 exploits the linear relationship between the reference pixels and the pixels to be predicted, TMP generally works better. However, for some blocks, mode 67 could not strike a good balance between complexity and performance, while mode 68 could achieve better results. Hence, mode 68 also made some contributions to video compression. For “BQMall” (832×480), the using probability of mode 68 reached 0.28%. These results illustrate that our proposed algorithm showed good compression efficiency.

To show the using probability more intuitively, we plotted the numbers being used in all luma modes of the sequence “BasketballPass” (416×240) in QP 32 and “BQTerrace” (1920×1080) in QP 27, as shown in Figure 8. The X-axis is the mode number, and the Y-axis is the number being used for each mode. We drew two dotted lines with the numbers being used for mode 67 and mode 68, respectively, to show the comparison of the results more intuitively. Here, the red dotted line represents the numbers of mode 67 being used, and the green one represents the numbers of mode 68 being used. If the blue “+” is below the red dotted line, it means that the using numbers of this mode are less than that of mode 67. Correspondingly, if the blue “+” is below the green dotted line, it means that the numbers being used of this mode are less than that of mode 68. In Figure 8b, there is no red dotted line. This is because mode 67 had been used more than 1200 times and could not be explicitly shown in the figure. Clearly, the numbers being used for mode 67 were much higher than that of most VVC traditional modes. Although the using probability of mode 68 was only 0.29% (“BasketballPass” in QP 32) and 0.22% (“BQTerrace” in QP 22), the numbers being used for mode 68 were still higher than some of the original VVC intra prediction modes.

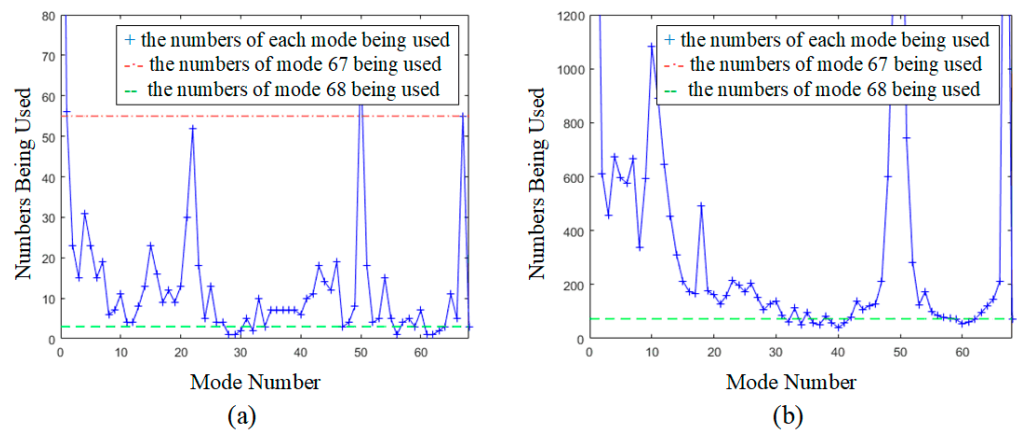


Figure 8. The numbers being used for each mode: (a) “BasketballPass” in QP 32; (b) “BQTerrace” in QP 22.

Table 5 shows the total using probability of mode 67 and mode 68 in the chroma component. For the chroma blocks, the optimal mode can be mode 67 or mode 68, only if that of the corresponding luminance block is mode 67 or mode 68 according to the VVC prediction process. Therefore, the total using probability of our proposed modes in the chroma component was lower than that of the luminance component. In the chroma component, there were also correlations between the different blocks and different modes, hence, our algorithm was still effective for the chroma components.

Table 5. The total using probability of mode 67 and mode 68 in the chroma component.

Sequence	QP = 22	QP = 27	QP = 32	QP = 37	Average
BasketballPass	4.81%	2.15%	1.35%	1.31%	2.41%
BQMall	0.42%	0.26%	0.36%	0.00%	0.26%
Johnny	3.58%	3.41%	2.79%	3.54%	3.33%
BasketballDrive	2.99%	3.72%	5.33%	3.89%	3.98%
BQTerrace	4.28%	4.65%	4.69%	3.16%	4.20%

In addition, we present the modes using subjective graphs of some sequences in QP 27. Figures 9–11 show the respective results. The red rectangular boxes are the blocks that utilized our proposed algorithm to conduct intra prediction. As can be seen, in some areas with similar textures, our modes were more competitive. This is because the traditional VVC intra prediction modes are more suitable for blocks with a single texture direction or simple areas, and our proposed algorithm can partly compensate for this shortcoming. These results further verify the good performance of our proposed fusion-based algorithm.



Figure 9. The mode subjective graphs. Images from the JVET CTC [42]: (a) “BQSquare” (416 × 240); (b) “BQMall” (832 × 480).

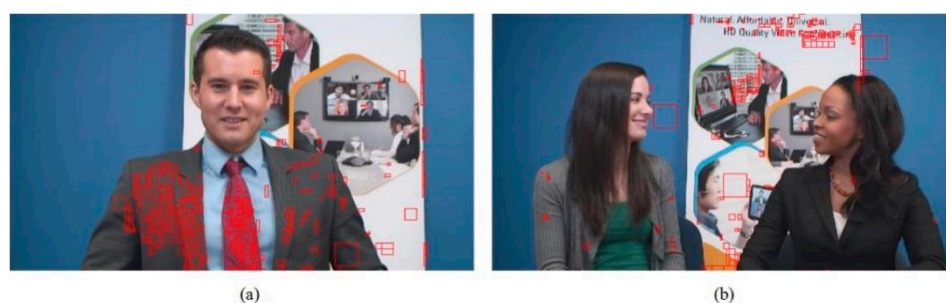


Figure 10. The mode subjective graphs. Images from the JVET CTC [42]: (a) “Johnny” (1280 × 720); (b) “KristenAndSara” (1280 × 720).

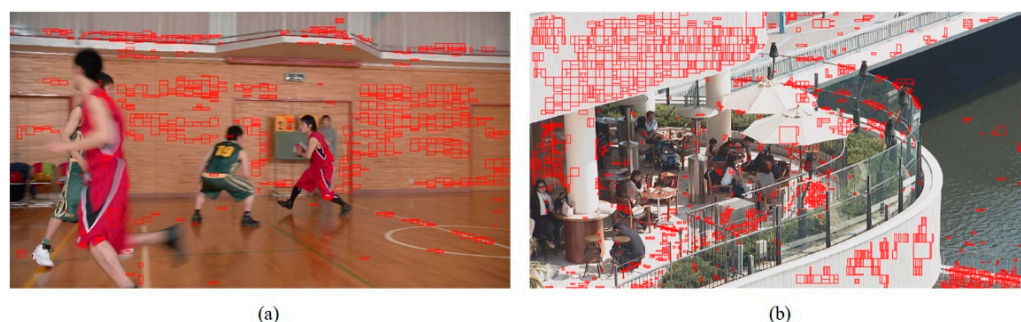


Figure 11. The mode subjective graphs. Images from the JVET CTC [42]: (a) “BasketballDrive” (1920 × 1080); (b) “BQTerrace” (1920 × 1080).

5. Conclusions

In this paper, we designed an efficient fusion-based algorithm to enhance the compression performance of VVC intra prediction. First, we presented a fusion-based adaptive template matching method to better predict areas with similar texture. The error between the reference template and candidate template was used to make the prediction results more precise. Second, we presented a fusion-based linear prediction method to fully utilize the correlation between the reference samples and the pixels to be predicted. This method also compensated for the shortcomings of the single linear prediction. Experiments verified the effectiveness of our proposed algorithm. Compared with VTM 9.1, the bitrate savings for components Y, Cb, and Cr achieved 0.89, 0.84, and 0.90% on average, respectively. The maximum bitrate savings were up to 2.69%, 2.81%, and 2.81% for Y, Cb, and Cr, respectively. In addition, the comparison with other works and using probability further verified that our proposed algorithm improved the intra prediction performance. For some classes, the bitrate saving was not that notable. This is mainly because the VVC itself has adopted many new techniques to optimize the performance of intra prediction, which makes it difficult to improve the performance. In the future, we will try to further improve the performance of intra prediction.

Author Contributions: Conceptualization, D.L.; Methodology, D.L.; Software, D.L.; Validation, D.L.; Formal analysis, D.L.; Investigation, D.L.; Writing—original draft preparation, D.L.; Writing—review and editing, C.R. and D.L.; Visualization, D.L.; Supervision, D.L., S.X., C.R., R.E.S., and X.H.; Project administration, C.R. and D.L.; Funding acquisition, S.X., C.R., R.E.S., and X.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by the National Natural Science Foundation of China (62211530110, 61871279, and 62171304).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Bross, B.; Chen, J.; Liu, S.; Wang, Y.K. Versatile video coding (draft 8), document JVET-Q2001. In Proceedings of the 17th JVET Meeting, Brussels, Belgium, 7–17 January 2020.
2. Zhu, X.; Liu, F.; Hu, D.H. 265/HEVC: Principle, Standard and Implementation. *Electron. Ind. Press* **2013**, *3*, 154–196.
3. Chen, J.; Ye, Y.; Kim, S.H. Algorithm description for versatile video coding and test model 7 (VTM 7), document JVET-P2002. In Proceedings of the 16th JVET Meeting, Geneva, Switzerland, 1–11 October 2019.
4. Zhao, X.; Seregin, V.; Said, A.; Zhang, K.; Egilmez, H.E.; Karczewicz, M. Low complexity intra prediction refinements for video coding. In Proceedings of the 2018 IEEE Picture Coding Symposium (PCS), San Francisco, CA, USA, 24–27 June 2018; pp. 139–143.
5. Said, A.; Zhao, X.; Karczewicz, M.; Chen, J.; Zou, F. Position dependent prediction combination for intra-frame coding. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 534–538.
6. Chang, Y.J.; Jhu, H.J.; Jiang, H.Y.; Zhao, L.; Liu, S.; Wiegand, T. 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.
7. Bross, B.; Keydel, P.; Schwarz, H.; Marpe, D.; Wiegand, T. Multiple reference line intra prediction, document JVET-L0283. In Proceedings of the 12th JVET Meeting, Macao, China, 3–12 October 2018.
8. De-Luxan-Hernandez, S.; George, V.; Ma, J.; Nguyen, T.; Schwarz, H.; Marpe, D.; Wiegand, T. An intra sub-partition 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.
9. De Luxan Hernandez, S.; Schwarz, H.; Marpe, D.; Wiegand, T. CE3: Line-based intra coding mode, document JVET-K0049. In Proceedings of the 11th JVET Meeting, Ljubljana, Slovenia, 10–18 July 2018.
10. Choi, J.; Heo, J.; Lim, J. Non-CE3: CCLM block size restriction, document JVET-O0399. In Proceedings of the 15th JVET Meeting, Gothenburg, Sweden, 3–12 July 2019.
11. Huo, J.; Ma, Y.; Wan, S.; Yu, Y.; Wang, M.; Zhang, K.; Zhang, L.; Liu, H.; Xu, J.; Wang, Y.; et al. CE3-1.5: CCLM derived with four neighbouring samples, document JVET-N0271. In Proceedings of the 14th JVET Meeting, Geneva, Switzerland, 19–27 March 2019.
12. Laroche, G.; Taquet, J.; Gisquet, C.; Onno, P. CE3-5.1: On cross-component linear model simplification, document JVET-L0191. In Proceedings of the 12th JVET Meeting, Macao, China, 3–12 October 2018.
13. Schafer, 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.
14. Schäfer, M.; Merkle, P.; Helle, P.; Hinz, T.; Schwarz, H.; Marpe, D.; Wiegand, T.; Pfaff, J.; Stallenberger, B. Affine linear weighted intra prediction, document JVET-N0217. In Proceedings of the 14th JVET Meeting, Geneva, Switzerland, 19–27 March 2019.
15. Zhao, L.; Zhao, X.; Liu, S.; Li, X.; Lainema, J.; Rath, G.; Urban, F.; Racape, F. Wide angular intra prediction for versatile video coding. In Proceedings of the 2019 Data Compression Conference (DCC), Snowbird, UT, USA, 26–29 March 2019; pp. 53–62.
16. Racapé, F.; Rath, G.; Urban, F.; Zhao, L.; Liu, S.; Zhao, X.; Li, X.; Filippov, A.; Rufitskiy, V.; Chen, J. CE3-related: Wide-angle intra prediction for non-square blocks, document JVET-K0500. In Proceedings of the 11th JVET meeting, Ljubljana, Slovenia, 10–18 July 2018.
17. Zhao, L.; Zhao, X.; Liu, S.; Li, X. CE3-related: Unification of angular intra prediction for square and non-square blocks, document JVET-L0279. In Proceedings of the 12th JVET Meeting, Macao, China, 3–12 October 2018.
18. Xiu, X.; He, Y.; Ye, Y. CE9-Related: Complexity reduction and bit-width control for bi-directional optical flow (BIO), document JVET-L0256. In Proceedings of the 12th JVET meeting, Macao, China, 3–12 October 2018.
19. Kato, Y.; Toma, T.; Abe, K. Simplification of BDOF, document JVET-O0304. In Proceedings of the 15th JVET Meeting, Gothenburg, Sweden, 3–12 July 2019.
20. Sethuraman, S. CE9: Results of DMVR Related Tests CE9.2.1 and CE9.2.2, document JVET-M0147. In Proceedings of the 13th JVET Meeting, Marrakech, Morocco, 9–18 January 2019.
21. Lin, S.; Chen, H.; Zhang, H.; Sychev, M.; Yang, H.; Zhou, J. Affine transform prediction for next generation video coding, document COM16-C1016. In *Huawei Technologies, International Organisation for Standardisation Organisation Internationale de Normalisation ISO/IEC JTC1/SC29/WG11 Coding of Moving Pictures and Audio, ISO/IEC JTC1/SC29/WG11 MPEG2015/m37525*; ISO: Geneva, Switzerland, 2015.
22. Chen, J.; Karczewicz, M.; Huang, Y.W.; Choi, K.; Ohm, J.R.; Sullivan, G.J. The joint exploration model (JEM) for video compression with capability beyond HEVC. *IEEE Trans. Circ. Syst. Video Technol.* **2020**, *30*, 1208–1225. [\[CrossRef\]](#)
23. Lainema, J. CE6-Related: Shape adaptive transform selection, document JVET-L0134. In Proceedings of the 12th JVET Meeting, Macao, China, 3–12 October 2018.
24. Zhang, C.; Ugur, K.; Lainema, J.; Hallapuro, A.; Gabbouj, M. Video coding using spatially varying transform. *IEEE Trans. Circ. Syst. Video Technol.* **2011**, *21*, 127–140. [\[CrossRef\]](#)
25. Koo, M.; Salehifar, M.; Lim, J.; Kim, S.H. Low frequency nonseparable transform (LFNST). In Proceedings of the 2019 Picture Coding Symposium (PCS), Ningbo, China, 12–15 November 2019; pp. 1–5.

26. Schneider, J.; Mehlem, D.; Meyer, M.; Rohlfing, C. Sparse coding-based intra prediction in VVC. In Proceedings of the 2021 Picture Coding Symposium (PCS), Virtual, 29 June–2 July 2021; pp. 1–5.
27. Li, C.; Zhao, Z.; Li, J.; Zhang, X.; Ma, S.; Li, C. Bi-intra prediction for versatile video coding. In Proceedings of the 2019 Data Compression Conference (DCC), Snowbird, UT, USA, 26–29 March 2019; p. 587.
28. Yoon, Y.U.; Park, D.H.; Kim, J.G. Enhanced derivation of model parameters for cross-component linear model (CCLM) in VVC. *IEEE Trans. Informat. Syst.* **2020**, *103*, 469–471. [\[CrossRef\]](#)
29. Zhu, W.; Xu, J.; Zhang, L.; Zhang, K.; Liu, H.; Wang, Y. Compound palette mode for screen content coding. In Proceedings of the 2019 Picture Coding Symposium (PCS), Ningbo, China, 12–15 November 2019; pp. 1–5.
30. Adhuran, J.; Kulupana, G.; Blasi, S.; Fernando, A. Parameter-based affine intra prediction of screen content in versatile video coding. *IEEE Trans. Circ. Syst. Video Technol.* **2021**, *31*, 3590–3602. [\[CrossRef\]](#)
31. Weng, X.; Lin, M.; Lin, Q.; Chen, G. L-shape-based iterative prediction and residual median edge detection (LIP-RMED) algorithm for VVC intra-frame lossless compression. *IEEE Signal Process. Lett.* **2022**, *29*, 1227–1231. [\[CrossRef\]](#)
32. Yoon, Y.U.; Park, D.H.; Kim, J.G.; Lee, J.; Kang, J. Most frequent mode for intra-mode coding in video coding. *Electron. Lett.* **2019**, *55*, 188–190. [\[CrossRef\]](#)
33. Wang, R.; Tang, L.; Tang, T. Fast sample adaptive offset jointly based on HOG features and depth information for VVC in visual sensor networks. *Sensors* **2020**, *20*, 6754. [\[CrossRef\]](#) [\[PubMed\]](#)
34. Saha, A.; Chavarrias, M.; Pescador, F.; Ángel, M.G.; Kheyter, C.; Pedro, L.C. Complexity analysis of a versatile video coding decoder over embedded systems and general purpose processors. *Sensors* **2021**, *21*, 3320. [\[CrossRef\]](#) [\[PubMed\]](#)
35. Tan, T.K.; Boon, C.S.; Suzuki, Y. Intra prediction by averaged template matching predictors. In Proceedings of the 2007 4th Consumer Communications and Networking Conference (CCNC), Las Vegas, NV, USA, 11–13 January 2007; pp. 405–409.
36. Venugopal, G.; Helle, P.; Mueller, K.; Marpe, D.; Wiegand, T. Hardware-friendly intra region-based template matching for VVC. In Proceedings of the 2019 Data Compression Conference (DCC), Snowbird, UT, USA, 26–29 March 2019; p. 606.
37. Venugopal, G.; Müller, K.; Marpe, D.; Wiegand, T.A. Unified region-based template matching approach for intra and inter prediction in VVC. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019; pp. 4115–4119.
38. Lei, M.; Luo, F.; Zhang, X.; Wang, S.; Ma, S. Two-step progressive intra prediction for versatile video coding. In Proceedings of the 2020 IEEE International Conference on Image Processing (ICIP), Virtual, 25–28 October 2020; pp. 1137–1141.
39. Ghaznavi-Youvalari, R.; Lainema, J. Joint cross-component linear model for chroma intra prediction. In Proceedings of the 2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSp), Virtual, 21–24 September 2020; pp. 1–5.
40. Zhang, K.; Chen, J.; Zhang, L.; Li, X.; Karczewicz, M. Enhanced cross-component linear model for chroma intra-prediction in video coding. *IEEE Trans. Image Process.* **2018**, *27*, 3983–3997. [\[CrossRef\]](#) [\[PubMed\]](#)
41. Ghaznavi-Youvalari, R. Linear model-based intra prediction in VVC test model. In Proceedings of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Virtual, 4–8 May 2020; pp. 4417–4421.
42. Boyce, J.; Suehring, K.; Li, X.; Seregin, V. JVET common test conditions and software reference configurations, document JVET-J1010. In Proceedings of the 10th JVET Meeting, San Diego, CA, USA, 10–20 April 2018.
43. Bjontegaard, G. Calculation of average PSNR differences between RD-Curves. In Proceedings of the 2001 ITU-T Video Coding Experts Group (VCEG) Thirteenth Meeting, Austin, TX, USA, 2–4 April 2001.