UNIFIED MATRIX CODING FOR NN ORIGINATED MIP IN H.266/VVC

Junyan Huo[*], *Yu Sun*[*], *Haixin Wang*[*], *Shuai Wan*[†], *Fuzheng Yang*[*], *Ming Li*[‡]

[*]State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, Shaanxi, China
[†]School of Electronics and Information, Northwestern Polytechnical University, Xi'an, Shaanxi, China
[‡]OPPO Mobile Telecommunications Corp., Ltd, Shenzhen, Guangdong, China

ABSTRACT

Matrix-based Intra Prediction (MIP) is an effective coding algorithm in H.266/Versatile Video Coding (VVC) which is originated by Neural Networks (NN). With the requirement of low complexity, MIP is conducted by a matrix-vector multiplication. To handle with the diversity of video content, 30 matrices are trained and stored to derive predicted samples. Since matrices from training are usually floating-point values, which should be avoided in H.266/VVC, two parameters, *shift* and *offset*, are introduced for each matrix to convert floating-point values to integers. This paper designs an efficient algorithm to determine the input vector of MIP, with which the range of the matrices can be minimized, and all matrices can be converted to integers with a unified *shift* and a unified *offset*. The proposed algorithm removes the matrix-dependent parameters for integer conversion and saves the memory for storing MIP parameters. Experimental results demonstrate that the proposed algorithm has a similar coding performance with VVC reference software. Due to the unified operation, memory reduction, and no coding loss, the proposed algorithm has been adopted into H.266/VVC.

Index Terms— Versatile Video Coding, H.266, matrix-based intra prediction, unified matrix coding

1. INTRODUCTION

H.266/VVC [1], published in 2020, is the latest video coding standard developed by Joint Video Experts Team (JVET). It brings significant coding performance improvement over H.265/HEVC [2], especially for high resolution videos. Besides, better supports to versatile applications are considered, such as screen content videos, 360-degree videos, and high-dynamic-range videos.

To achieve higher coding performance, many excellent coding algorithms are newly introduced into each coding module of H.266/VVC [3]. As for the prediction coding module, coding tools, such as Matrix-based Intra Prediction (MIP), Cross-Component Linear Model (CCLM), and Geometric Partitioning Mode (GPM), focus on improving the accuracy of predicted samples.

Different from the traditional coding tools, MIP is a data-driven algorithm in which the matrices in MIP are originated by Neural Networks (NN). The first version of MIP, NN-based Intra Prediction (NNIP) [4], employed a four-layer fully connected network to obtain the predicted samples. Such a scheme can provide an excellent coding gain benefitted from the nonlinearity of NN. However, since both the computational complexity and the memory have a strict restrict in H.266/VVC, several algorithms were designed to reduce the complexity of NNIP. Finally, MIP reaches a good trade-off between the coding performance and the complexity and has been adopted by H.266/VVC Draft 5 [5].

When incorporating MIP into H.266/VVC, to provide better prediction for diverse video content, 30 matrices are introduced for different block sizes. The weights of matrices are usually floating-point values. A key issue to incorporate MIP into H.266/VVC is integer conversion. In H.266/VVC Draft 5, two parameters, *shift* and *offset*, are introduced for each matrix to convert floating-point values to integers.

In this paper, we propose a unified *shift* and *offset* scheme based on a deep analysis of MIP weights. Specifically, the distribution of MIP weights is analyzed first. Based on that, the input vector of MIP is improved, with which the range of weights is more concentrate. Finally, a unified integer conversion scheme is proposed. Due to its unification, memory reduction, and no coding loss, the proposed scheme has been adopted into H.266/VVC [1].

The rest of the paper is organized as follows. The NNIP and MIP in H.266/VVC is reviewed in Section 2. The proposed unified integer conversion scheme is presented in Section 3. Experimental results and analysis are given in Section 4. Finally, conclusions are drawn in Section 5.

2. FROM NN-BASED INTRA PREDICTION TO MIP

2.1. NN-based Intra Prediction

NNIP proposed some non-linear intra prediction modes to generate the predicted samples of a block. These intra prediction modes perform the following two main steps: First, a set of features is extracted from the decoded neighboring samples. Second, these features are used to select an affine

linear combination of predefined image patterns as the predicted samples.

In the paper proposed by Pfaff *et al.* [4], NNIP has multiple network layers and multiple modes for intra prediction. In order to reduce the complexity of computing and storage and to achieve hardware-friendliness, Helle *et al.* [6] proposed a new network in which the number of layers was reduced to two and the number of modes was halved. In order to further reduce the complexity, a new algorithm [7] based on NNIP was proposed which simplified the calculation method of input vector and carried out twice interpolation up-sampling of the prediction results. In order to reduce the input dimension and obtain better prediction results, the input of the network was converted from the pixel domain to the frequency domain [8].

With lots of simplification efforts, NNIP has been officially adopted in H.266/VVC as MIP. It only uses one layer of fully connected network training weights, which can be stored in memory in the form of some integer matrices pre-stored table. Adopted NNIP takes down-sampling of reference pixels, matrix-vector multiplication, and up-sampling of the matrix operation output as the key steps, which greatly reduces the complexity of NNIP, and is also renamed as matrix-based intra prediction, abbreviated as MIP.

2.2. MIP in H.266/VVC

The prediction derivation of MIP [7] is illustrated in Fig. 1, where the predicted samples are obtained through matrix multiplication and offset addition. Specifically, the prediction vector, $\boldsymbol{P}$, is represented as

$$\boldsymbol{P} = \boldsymbol{w} \cdot \boldsymbol{D} + Y'_0, \tag{1}$$

where $\boldsymbol{w}$ is the weight matrix obtained from extensive offline data training, $\boldsymbol{D}$ is a vector calculated from the neighbouring vector $\boldsymbol{Y_N}$, $\boldsymbol{Y'_N}$ is the down-sampled neighbouring vector and $\boldsymbol{Y'_0}$ is the first element of $\boldsymbol{Y'_N}$.

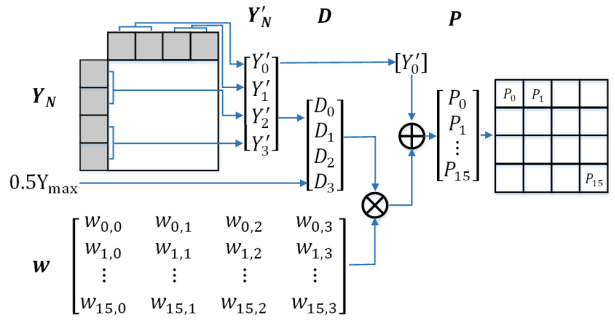

Fig. 1. The prediction derivation of MIP

The vector $\boldsymbol{D}$ in H.266/VVC Draft 5 is designed according to the block size. For coding blocks with width and height larger than 4 (except for 8×8 blocks), $\boldsymbol{D}$ is derived as

$$D_j = Y'_{j+1} - Y'_0. \quad j = 0, 1, ..., size(\boldsymbol{Y'_N}) - 2 \tag{2}$$

Here, D_j is the difference between the down-sampled neighbouring element Y'_{j+1} and Y'_0. For coding blocks with a small block size, i.e. 8×8 blocks or blocks with the width or height of 4, $\boldsymbol{D}$ is designed as

$$D_0 = Y'_0 - 0.5Y_{\max}, j = 0 \tag{3}$$

$$D_j = Y'_j - Y'_0, j = 1, ..., size(\boldsymbol{Y'_N}) - 1 \tag{4}$$

where the derivation of D_j in (4) is similar as that in (2), D_0 is the difference between Y'_0 and $0.5Y_{\max}$. Here, $0.5Y_{\max}$ is equal to $1 << (bitDepth - 1)$ and *bitDepth* is the bit depth of luma samples.

During NN training, elements in $\boldsymbol{w}$ are usually floating-point values. While, floating-point operation needs to consume lots of computational resource which is not desirable in video coding application. In H.266/VVC, a floating-point free process is conducted and the integer weight matrix $\boldsymbol{W}$ is derived from $\boldsymbol{w}$. Firstly, each element in $\boldsymbol{w}$ is multiplied with $1 << shift$ and then rounded into integers. Secondly, these integers add with *offset* to ensure that elements in $\boldsymbol{W}$ are non-negative integers.

In summarize, the MIP predicted samples in H.266/VVC Draft 5 are designed as

$$P_i = \left(\left(\sum\nolimits_j (W_{i,j} - offset) \cdot D_j \right) >> shift \right) + Y'_0 \tag{5}$$

To meet the diversity of video content, 30 matrices are employed in MIP to construct the predicted samples. Since the ranges of these matrices are different, the matrix-dependent *shift* and *offset* are introduced in Table 1 and Table 2.

Table 1. Matrix-dependent *shift*

index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	*6*	*5*	*5*	*5*	*5*	*6*	*6*	*6*	*6*	*6*	*6*	*5*	*6*	*5*	*5*	*5*
1	*7*	*7*	*6*	*6*	*6*	*6*	*6*	*6*								
2	*6*	*7*	*5*	*6*	*6*	*6*										

Table 2. Matrix-dependent *offset*

index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	*1*	*28*	*28*	*42*	*56*	*22*	*9*	*6*	*35*	*14*	*50*	*66*	*29*	*50*	*31*	*19*
1	*15*	*14*	*23*	*45*	*10*	*14*	*10*	*27*								
2	*15*	*19*	*46*	*16*	*14*	*11*										

3. PROPOSED MIP ALGORITHM WITH UNIFIED MATRIX CODING

Since the predicted samples are generated in the encoder and decoder, weight matrices in MIP need to be stored. In H.266/VVC Draft 5, two look-up-tables, including 30 *shifts*

and *offsets*, also need to be stored in order to derive the predicted samples. In this section, the MIP prediction derivation is further improved and a unified *shift* and *offset* algorithm is proposed [9]. Due to its simplification, memory reduction and no coding loss, the proposed algorithm has been adopted into H.266/VVC [1].

In this section, the distribution of MIP weights in H.266 /VVC Draft 5 is first analyzed. Then, the input vector of MIP is improved to minimize the range of MIP weight. Based on the improved input vector, the MIP with unified *shift* and unified *offset* is proposed.

3.1. The distribution of MIP weights

According to the integer weight matrix, a reconstructed floating weight, $\hat{w}_{i,j}$ can be obtained as

$$\hat{w}_{i,j} = (W_{i,j} - offset) >> shift. \tag{6}$$

In H.266/VVC Draft 5, the elements of $\boldsymbol{W}$ are located within [0,127]. Therefore, elements of $\hat{\boldsymbol{w}}$ are in the range of

$$\hat{w}_{i,j} \in \left(\frac{-offset - 0.5}{2^{shift}}, \frac{127 - offset + 0.5}{2^{shift}} \right] \tag{7}$$

where different *shifts* and *offsets* lead to different range of $\hat{\boldsymbol{w}}$. The distribution of 30 weight matrices in H.266/VVC Draft 5 is provided in Fig. 2 where the x-axis represents the intensity of $\hat{\boldsymbol{w}}$ and the y-axis shows the probability of $\hat{\boldsymbol{w}}$. From Fig. 2, we can see that most of weights are around zero. As for the negative weights, the probability is far less than that of non-negative values.

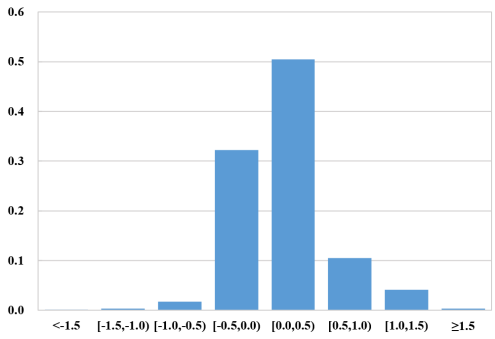

Fig. 2. Weight distribution in H.266/VVC

Fig. 3 further lists each weight matrix of H.266/VVC Draft 5. The x-axis represents the index of weight matrices, and the y-axis shows all elements of $\hat{\boldsymbol{w}}$. We can see that there are a few negative weights whose absolute intensities are larger than 1.5. Because of these small amounts of negative weights with large absolute intensities, the *shift* and *offset* of each weight matrix need to be recorded separately to ensure the precision of each weight matrix.

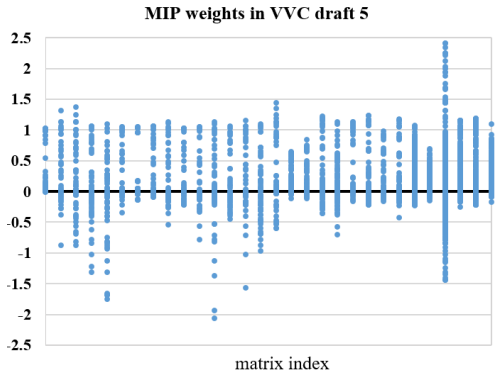

Fig. 3. MIP weight in H.266/VVC Draft 5

In order to design a unified *shift* and *offset* and maintain the floating-point precision, the distribution of $\hat{\boldsymbol{w}}$ need to concentrate into a small range. Based on this motivation, we analyse the range distribution in Fig. 3 and find that most of these weights with large absolute intensities are from the first entry of $\hat{\boldsymbol{w}}$ which is the corresponding weight of D_0.

3.2. Improved input vector calculation

In order to derive weight matrices with a mall range, the proposed D_0 is designed as

$$D_j = 0.5Y_{\max} - Y'_0, \quad j = 0 \tag{8}$$

The reason for this design is as follows. In H.266/VVC, when a block is coded with a small size, it is usually considered as a textural block. The samples of such a block are hard to predict by using the neighbouring vector, especially for those pixels far from the neighbouring region. Therefore, $0.5Y_{\max}$, the expectation of luma under a uniform distribution assumption, is introduced as a candidate of the sample.

We further propose to calculate the vector $\boldsymbol{D}$ as

$$D_j = Y'_{j+1} - Y'_0, j = 0, ..., size(\boldsymbol{Y'_N}) - 2 \tag{9}$$

$$D_j = 0.5Y_{\max} - Y'_0, j = size(\boldsymbol{Y'_N}) - 1 \tag{10}$$

where (9) can be applied to all blocks with different sizes and (10) is added as an efficient candidate for small blocks. Such a scheme can provide a unified calculation for blocks with different sizes. According to the definition of (1), the MIP prediction can be composed with a DC plus AC design. Y'_0 in (1) can be regarded as the DC part and the result of matrix multiplication can be regarded as the AC part. The reduction of Y'_0 to derive the vector $\boldsymbol{D}$ in (9) and (10) can be considered as the DC remove operation. During the data training, since the input signal is the AC part and independent of the video content, it is easier to optimize the matrix training. Such an algorithm is easy for converge.

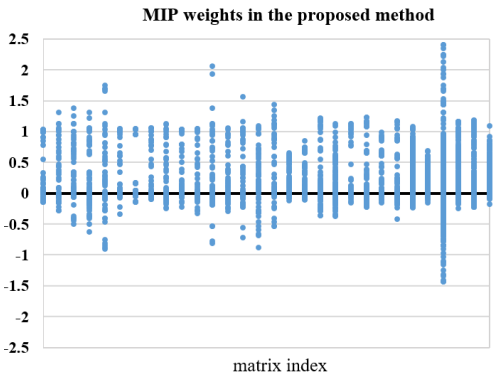

Fig. 4. MIP weight in the proposed algorithm

3.3. Unified MIP parameters design

Fig. 4 further lists the MIP weight of the proposed algorithm. We can see that the weight range of the proposed algorithm turns to be narrow which is located within -1.5 and 2.5. Such a range is benefit for the unified integer matrix design. Because the algorithm proposed in Section 3.2 effectively reduces the range of the matrix, we can represent the floating-point weight matrix with higher accuracy by increasing the number of *shift* from 5 to 6 without coding loss.

Based on the proposed algorithm, we set *shift* to 6 and *offset* to 32. The MIP prediction vector, $\boldsymbol{P}$, in the proposed algorithm is derived as

$$P_i = \left(\left(\sum\nolimits_j (W_{i,j} - 32) \cdot D_j \right) >> 6 \right) + Y_0'. \qquad (11)$$

Compared with (5), the calculation of (11) can be regarded as a simplification of MIP since it removes the dependency of *shift* and *offset* on matrix.

4. EXPERIMENTAL RESULTS AND ANALYSIS

To verify the coding performance, the proposed algorithm is integrated based on H.266/VVC reference software, VVC Test Model (VTM) version 7.0 [10]. The test configurations are performed according to the Common Test Condition (CTC) as specified in [11]. The experiments are conducted on an Intel Xeon cluster (E5-2697A v4, AVX2 on, turbo boost off) with Linux OS and GCC 7.2.1 compiler. To evaluate the overall coding performance, 7 classes of test sequences, covering different video contents and different video resolutions are employed. Bjontegaared-Delta bitrate (BD-rate) [12] is used to evaluate the objective rate-distortion performance. To obtain four rate points, the experiments are performed with a quantization parameter (QP) of 22, 27, 32, and 37.

Table 3 shows the performance results of the proposed algorithm for each class in All Intra (AI) configuration. Compared to VTM7.0, the proposed algorithm brings -0.01%, 0.01% and 0.04% BD-rate on average for Y, Cb, and Cr components, respectively, with 99% encoding time and 99% decoding time. Form the data, we can see that the proposed algorithm has a similar coding performance with VTM 7.0 and the encoding and decoding time are slightly reduced. The reduction in codec time is due to the removal of two look-up tables, and the savings in BD-rate are mainly due to the improvement in *shift* accuracy.

Table 3. BD-rate results of the proposed algorithm over VTM7.0 for Each Class in AI configuration

Class	Y	Cb	Cr	enctime	dectime
A1	-0.02%	-0.19%	0.15%	100%	103%
A2	0.00%	0.01%	-0.03%	99%	99%
B	0.01%	0.00%	-0.05%	99%	97%
C	0.00%	-0.19%	0.02%	97%	99%
D	-0.02%	0.09%	0.14%	101%	99%
E	-0.04%	0.16%	0.05%	100%	100%
F	-0.02%	-0.03%	0.05%	99%	99%
Avg	**-0.01%**	**-0.19%**	**0.04%**	**99%**	**99%**

Notably, using a unified *shift* and a unified *offset* to replace the matrix-depend parameters, if the unified parameters are not properly designed, generally means sacrifices in coding performance. In this paper, such unified parameters design is achieved by improving the derivation of the input vector. With the proposed derivation minimizing the range of the matrices, a unified integer conversion algorithm can be easily used. As a result, the derivation of (8) is crucial for the unified design.

One benefit of the proposed algorithm is to save the storage requirement of MIP. In the proposed algorithm, the dependency between the *shift* and *offset* with the matrix are removed. In this way, for MIP which needs to store matrix parameters, 300 bits can be saved, including 90 bits of *shift* and 210 bits of *offset*. That is to say, when the prediction process of MIP is conducted, there is no need to access the memory to fetch the *shift* and *offset* for each MIP mode.

More importantly, by using the proposed unified *shift* and *offset* algorithm, the calculation of the MIP prediction can be unified for different block sizes and less look-up table operation is needed, a feature friendly to hardware implementation.

5. CONCLUSION

In this paper, an unified matrix coding algorithm in the matrix multiplication of MIP is proposed, removing the dependency of *shift* and *offset* on the index of matrix and saving the MIP parameters memory. Experimental results of -0.01% BD-rate change for the AI configuration are reported. The measured encoding- and decoding-times are 99% and 99%. The algorithm has been adopted by H.266/VVC and integrated into VTM version 8.0 due to its unification, memory reduction, and no coding loss.

6. REFERENCES

[1] "Versatile Video Coding", Standard Rec. ITU-T H.266, ISO/IEC 23090-3 VVC, Aug. 2020.

[2] High Efficiency Video Coding, Version 1, Standard Rec. ITU-T H.265, ISO/IEC 23008-2, Jan. 2013.

[3] B. Bross, J. Chen, J. R. Ohm, G. J. Sullivan, Y. Wang, "Developments in international video coding standardization after AVC, with an overview of Versatile Video Coding (VVC)", Proceedings of the IEEE, early access.

[4] J. Pfaff, H. Schwarz, D. Marpe, B. Bross, S. De-Luxán-Hernández, P. Helle, C. Helmrich, T. Hinz, W. Lim, J. Ma, T. Nguyen, J. Rasch, M. Schafer, M. Siekmann, G. Venugopal, A. Wieckowski, M. Winken, and T. Wiegand, "Video compression using generalized binary partitioning, trellis coded quantization, perceptually optimized encoding, and advanced prediction and transform coding", IEEE Transactions on Circuits and Systems for Video Technology, May 2020, 30, pp. 1281-1295.

[5] B. Bross, J. Chen, S. Liu, "Versatile Video Coding (Draft 5)", JVET-N1001, Mar. 2019.

[6] P. Helle, J. Pfaff, M. Schafer, R. Rischke, T. Wiegand, "Intra picture prediction for video coding with neural networks", Data Compression Conference (DCC), Mar. 2019.

[7] M. Schafer, B. Stallenberger, J. Pfaff, P. Helle, H. Schwarz, D. Marpe, T. Wiegand, "An affine-linear intra prediction with complexity constraints", IEEE International Conference on Image Processing (ICIP), Sept. 2019.

[8] M. Schafer, B. Stallenberger, J. Pfaff, ¨P. Helle, H. Schwarz, D. Marpe, T. Wiegand, "A data-trained, affine-linear intra-picture prediction in the frequency domain", Picture Coding Symposium (PCS), Nov. 2019.

[9] J. Pfaff, B. Stallenberger, M. Schafer, P. Merkle, P. Helle, T. Hinz, H. Schwarz, D. Marpe, T. Wiegand, K. Kondo, M. Ikeda, J. Huo, H. Wang, Y. Ma, F. Yang, S. Wan, Y. Yu, "MIP with constant shifts and offsets", JVET-Q0446, Jan. 2020.

[10] https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftwareVTM/tags/VTM-7.0.

[11] F. Bossen, J. Boyce, X. Li, V. Seregin, K. Sühring., "JVET common test conditions and software reference configurations for SDR video", JVET-N1010, Mar. 2019.

[12] G. Bjontegaard, "Calculation of average PSNR differences between RD-Curves", VCEG-M33, Apr. 2001.