National Chiao Tung University

EECS International Graduate Program

Master's Thesis

用於 HEVC 可調視訊編碼中估測模式相依之像素權重畫面內預測演算法

MODE-DEPENDENT PIXEL-BASED WEIGHTED INTRA PREDICTION FOR HEVC SCALABLE EXTENSION

Student: Nguyen Tang Kha Duy

Advisors: Prof. Chung-Ping Chung

Prof. Wen-Hsiao Peng

August, 2013

用於 HEVC 可調視訊編碼中估測模式相依之像素權重畫面內預測演算法

Mode-Dependent Pixel-Based Weighted Intra Prediction for HEVC Scalable Extension

研 究 生： 阮繒可維　　Student: Nguyen Tang Kha Duy
指導教授： 鍾崇斌　　Advisors: Chung-Ping Chung
彭文孝　　Wen-Hsiao Peng

國立交通大學

電機資訊國際學位學程

碩士論文

A Thesis

Submitted to EECS International Graduate Program

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

In

Electrical Engineering and Computer Science

August 2013

Hsinchu, Taiwan, Republic of China

中華民國一百零二年八月

用於 HEVC 可調視訊編碼中估測模式相依之像素權重畫面內預測演算法

研究生：阮繒可維　　　　　　　　指導教授：鍾崇斌

彭文孝

國立交通大學電機資訊國際學位學程

摘要

ISO/IEC 運動圖像專家組和 ITU-T 的視頻編碼專家組目前正在聯合開發 HEVC 的可伸縮性延長，他們提供了 RefIdx 和 TextureRL 這兩種方法來執行層間預測。根據 TextureRL 的設計，我們提出了一種模式相關的基於像素的加權幀內預測（MPWIP）編碼的增強層（EL）的方法。此方案首先分解的 EL 預測，通過常規的預測，與並列配置的基本層（BL）構建重構到它們各自的直流分量和交流分量的塊，然後使用一個基於像素的加權方案來計算它們的加權總和，以達到更好的預測信號。用最小二乘法擬合對數據進行訓練，得到與不同的組件相關聯的加權因子。根據觀察，在常用的測試條件下，他們強烈依賴 EL 的幀內預測模式和預測塊的大小，但不依賴於 QP 設置。相較於 SHM-1.0，MPWIP 在 AI-2 倍的配置下節省了約 1.0%的平均 BD 率，在 AI-1.5 的配置下節省了約 0.5%的平均 BD 率。此外，跟其它現有工程中表現最優秀的比較起來，在 AI-2 倍的配置下節省了約 0.3%的平均 BD 率，在 AI-1.5 的配置下節省了約 0.1%的平均 BD 率。

Mode-Dependent Pixel-Based Weighted Intra Prediction for HEVC Scalable Extension

Student: Nguyen Tang Kha Duy Advisors: Dr. Chung-Ping Chung
Dr. Wen-Hsiao Peng

EECS International Graduate Program
National Chiao Tung University

ABSTRACTION

A scalability extension to HEVC is currently under joint development of the ISO/IEC Moving Picture Experts Group and the ITU-T Video Coding Experts Group, offering two approaches, RefIdx and TextureRL, to performing inter-layer prediction. In the context of the TextureRL approach, this work proposes a mode-dependent pixel-based weighted intra prediction (MPWIP) for coding the enhancement layer (EL). The scheme first decomposes the EL predictor, constructed by the conventional directional prediction, and the collocated base-layer (BL) reconstructed block into their respective DC and AC components and then computes a weighted sum of both to form a better prediction signal using a pixel-based weighting scheme. The weighting factors to associate with different components are obtained by a least-squares fit to training data. It was observed that they depend strongly on the EL's intra prediction mode and prediction block size, but are less dependent on QP settings under the common test conditions. The MPWIP offers an average BD-rate savings of 1.0% for the AI-2x configuration and 0.5% for AI-1.5x over the SHM-1.0 anchor, as compared to 0.3% and 0.1% to the max shown for each configuration respectively in the other prior works.

Acknowledgement

Upon the completion of this Thesis, I would especially like to thank my advisors Professor Chung-Ping Chung and Professor Wen-Hsiao Peng for their academic guidance during the two years of my Master's program.

Professor Chung introduced me to Video Coding by providing the opportunity to enroll in a lab project related to scalable video coding. He also offered me the chance to work with the most knowledgeable professor in the field in our department, Professor Peng.

Through the kindness of Professor Peng, I was afforded the perfect opportunity to work in one of the most active research groups for Video Coding in Taiwan, the Multimedia Architecture and Processing Laboratory (MAPL). Weekly meetings and precise comments on each running MPEG meeting provided me with up-to date knowledge in SHVC from its beginning stages.

I also would give special thanks to Chun-Chi, a PhD student of the MALP lab for his comments on the simulation set up and checking for errors in this Thesis.

Finally, I want to thank my mentoring parents Pastor John Tolbert and Mrs. Beverly Tolbert. Special offers from them for mentoring and academic writing helps led me to finish this Thesis successfully.

Contents

List of Tables

List of Figures

CHAPTER 1

Research Overview

1.1. Introduction

One widely used approach, also known as the layered approach, for realizing scalable video coding (SVC) is to encode a video sequence into a base layer (BL) and one or more enhancement layers (ELs). The BL contains the most fundamental information to ensure a minimum guaranteed decoded quality, while the ELs produce improved quality when combined with the BL for decoding. Specifically, the more layers the decoder decodes, the better the decoded video quality (in terms of frame rate, resolution and reconstruction SNR).

Since the input video for all layers are from the same content, there is a strong correlation between the EL and the BL. Therefore, the number of bits required to code the EL can be significantly reduced by exploiting that correlation between the layers. As an example, in H.264/SVC, the redundancies between layers are reduced by employing various '*inter-layer prediction*' (ILP) mechanisms, such as inter-layer texture prediction (Intra-BL), inter-layer residual prediction, and inter-layer motion prediction.

Nowadays, the emergence of high quality content such as Ultra-High Definition (UHD) will likely change at an exponentially fast pace; therefore, as groundwork for future technology, the ITU-T and ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC) kicked off a state-of-the-art video coding standard, called High Efficiency Video Coding (HEVC). Currently, adding a scalable extension to HEVC (called SHVC) to take advantage of HEVC's improved video coding efficiency is under development.

In the present test model of SHVC, the ILP mechanisms can be realized by two different approaches, the RefIdx and the TextureRL [5]. In the RefIdx approach, the BL reconstructed picture is included in the reference picture lists, so that the inter-layer motion or texture prediction can be achieved by simply using the existing inter-frame prediction mechanisms. This approach has the benefit of being able to reuse most of the single layer HEVC design, except few high-level (e.g. slice level) syntax changes, for encoding and decoding the EL. On the contrary, the TextureRL approach allows to be changed the low-level (e.g. CU- and/or PU-level) syntax and decoding process, in hopes of getting a higher coding gain. Without being constrained to use inter-frame prediction mechanisms, this approach has more flexibility in forming inter-layer prediction. For example, a new prediction mode may be created at the PU level to better utilize both the BL and EL information for prediction. The price paid for this flexibility is however an EL codec that is less compatible with HEVC.

It was shown that the RefIdx approach can already achieve most of the gain provided by the TextureRL [13][14], making the latter a less favorable option. But, many believe the flexibility of the TextureRL has not been employed to the best advantage. Aiming to explore the potential of the TextureRL approach in forming a better inter-layer prediction, we propose in this work a mode-dependent pixel-based weighted intra prediction scheme for coding the EL.

1.2. Problem Statements

In the proposed scheme, we make a combined use of both the BL and EL information to create a better intra predictor. Specifically, we first decompose the EL intra predictor (created from the conventional directional prediction) and its collocated BL block into their respective DC[1] and AC[2] components. We then compute a weighted sum of these components using a mode-dependent pixel-based weighting scheme to form the final intra predictor for the EL.

The ideas are essentially a combination of two previously proposed notions in [6] and [7], where the former [6] introduces a *Weighted Intra Prediction* (WIP) scheme to linearly combine the EL intra predictor with the BL reconstructed block based on a pixel-based weighting scheme, while the latter [7] devises the notion of *Intra DC Correction* (IDCC) to improve the EL intra predictor by replacing its mean value (referred hereafter to as DC value) with that of the collocated block in the BL, which is motivated by the observation that the collocated BL block often gives a better estimate of the DC value of the input block than the EL intra predictor. Each of these

[1] The average sample values of a predictor

[2] The residual signal resulting from subtracting the DC component from a predictor

approaches has its own merits and faults. We attempt to combine their merits to form a prediction scheme that can improve further on prediction efficiency.

The combination of these notions is however not as easy as might be expected. Significant efforts, along with extensive experiments, have been made to:

- Determine how different components should be weighted.
- Determine how their weighting should be adapted according to the change in coding parameters, such as prediction mode, prediction block size, and QP setting.
- Strike a good balance between coding performance and complexity.

1.3. Contributions

Our main contributions in this work include:

- A mode-dependent pixel-based weighting scheme for combining DC and AC components of the BL reconstructed block and the EL intra predictor to improve intra prediction performance in the EL,
- An in-depth analysis of the combined effect of intra prediction mode, QP setting, and prediction block size on the weighting of different components,
- Simplifications to make the proposed algorithm practical and feasible.

Experimental results are provided under the All-Intra (AI) common test conditions for the two-layer cases having respectively a spatial scalability factor of 2x and 1.5x. In each of these test cases, our algorithm, when implemented into the SHM-1.0, shows an average BD-rate saving of 1.0% and 0.5%, respectively. The gains achieved are also observed to be higher than those provided by the two prior works, IDCC [7] and WIP [6].

1.4. Organization

The remainder of this thesis is organized as follows. Chapter 2 provides a brief review of SHVC and related prior works. Chapter 3 presents our proposed algorithm, including its concept of operation, the weighting schemes for different components, and how their weighting should be adapted in response to the change in coding parameters. Chapter 4 provides experimental results by implementing the proposed scheme with the SHM-1.0 software and conducting tests under the common test conditions [8]. Results for simplified schemes are also presented. Finally, Chapter 5 concludes this work with a summary of our findings and a list of future works.

CHAPTER 2

Background

2.1. H.264/AVC Scalable Video Coding

Scalable video coding (SVC) is aimed at generating a bit stream that allows part of it to be removed in such a way that the resulting sub-stream forms another valid bit stream and can still be decoded [10]. With scalability, users can experience graceful degradation of video quality rather than visible errors or interruptions. SVC supports three types of scalabilities: spatial, temporal and quality scalabilities.

- The **temporal scalability** is provided by utilizing the *hierarchical temporal prediction structure* of the single-layer coding [12].
- The **spatial scalability** makes use of a multi-layer pyramid approach in which separate encoding loops are employed for different spatial resolution layers and an adaptive inter-layer prediction mechanism is developed to exploit the correlations between layers.

- And lastly**, quality scalability** supports different quality levels with varying quantization parameters for different quality layers at the same spatial resolution.

In H.264/AVC video coding standard, the coded video data is organized into Network Abstraction Layer (NAL) units, which are packets that each contains an integer number of bytes. A set of NAL units in a specified form is referred to as an access unit. The decoding of each access unit results in one decoded picture. For an SVC bit stream, the set of corresponding access units can be partitioned into a base layer (BL) and one or more enhancement layers (ELs) with the following property. For each access unit, first the coding parameters of the BL are determined, and given these data, the ELs are coded. In SVC, the BL and the EL bit stream are coded by separate encoding loops, which are referred as coding layers; and in each layer, the basic concepts of inter-frame and intra-frame predictions are employed as in H.264/AVC. The correlation between layers is exploited by the Inter-Layer Prediction (ILP) mechanisms which are considered as additional coding options for the EL in the operational encoder control.

A simplification used throughout the discussions for this thesis is to primarily consider a bit stream containing only two layers (e.g. a lower resolution BL and a higher resolution spatial scalability EL). In fact, the SVC design fully support multilayer scenarios including multiple spatial scalability layers and the mixing of spatial scalability layers with other layers that provide temporal or quality scalability.

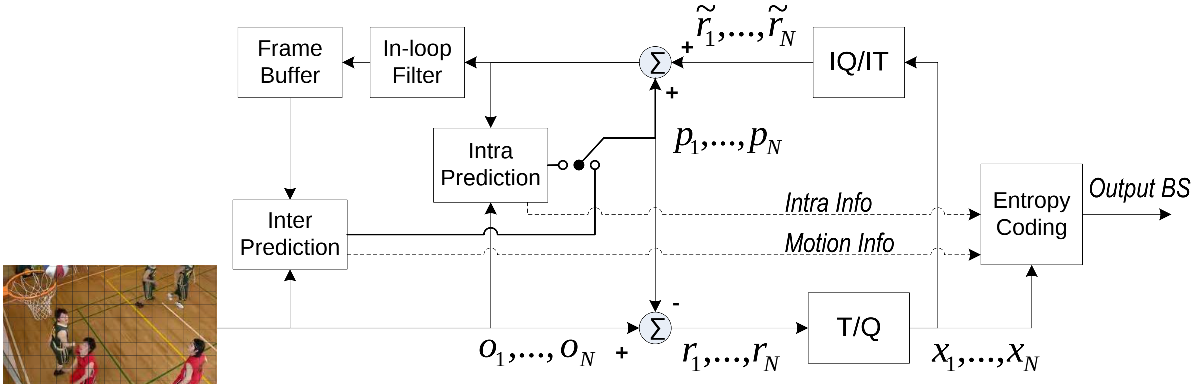


Figure 1: A high-level block diagram of an HEVC encoder

2.2. HEVC and the Scalable Extension to HEVC

In the scope of this thesis, the scalable video coding is an extension of High Efficiency Video Coding (HEVC), so-called Scalable High Efficiency Video Coding (SHVC). Therefore, a brief review on the coding structure of HEVC, including its similarities and differences with its predecessor H.264/AVC, will be introduced. In addition, the current status of adding scalable extension to HEVC will be analyzed.

2.2.1. Coding Structure of the HEVC

HEVC is a block-based hybrid video coding method which uses the spatial or temporal prediction, followed by transformation. In HEVC, an input frame is first divided into multiple coding tree units (CTUs), which are analogue of macroblock definition in H.264/AVC and other previous standard; but unlike macroblocks, CTUs have a variable size (e.g. 16x16, 32x32, or 64x64 samples). A CTU may contain only one CU (coding unit) or may be split to form multiple CUs, and every sample in a CU is either coded using inter-frame or intra-frame prediction. Also, CUs are divided into one or more multiple prediction units (PUs) and a tree of transform units (TUs). Inside a PU, the same prediction process (e.g. the same intra prediction mode, or the same motion information) is applied to all pixels.

Figure 1 shows a high-level block diagram of an HEVC encoder. For each PU, the residual pixels $r_1, \dots, r_N$ are generated by subtracting the predicted pixels $p_1, \dots, p_N$ (generated by either the *'Intra Prediction'* or *'Inter Prediction'* module) from the input $o_1, \dots, o_N$ pixels. Then, those residual values are fed into the *'T/Q'* module to form the quantized transform coefficients $x_1, \dots, x_N$, After that, entropy coding is performed on $x_1, \dots, x_N$ and side information, including intra mode and motion parameters, to generate the coded output bit stream. To ensure both the encoder and decoder will generate identical predictions for the subsequent data; the encoder duplicates the decoder processing loop inside. As a result, the reconstructed residual $\tilde{r}_1, \dots, \tilde{r}_N$ is generated by inverse scaling and then undergoes an inverse transformation processes inside the *'IQ/IT'* module. The '*In-loop Filter*' module is similar to the de-blocking filter module in H.264, but in addition it contains Sample Adaptive Offset module [3].

2.2.2. Coding Structure of the SHVC

Due to the fact that temporal scalability can generally be achieved by single-layer coding structure, the JCT-VC mainly focuses on developing scalable features to support spatial and coarse grain SNR scalabilities. Conventionally, in order to exploit the correlations between layers, the *Inter Layer Prediction* (ILP) mechanism is utilized. The ILP employs the coded data, including reconstructed pictures, intra mode, and motion parameters, in the reference layer for effectively coding an enhancement layer. A general block diagram of a two-layer SHVC encoder is depicted in Figure 2.

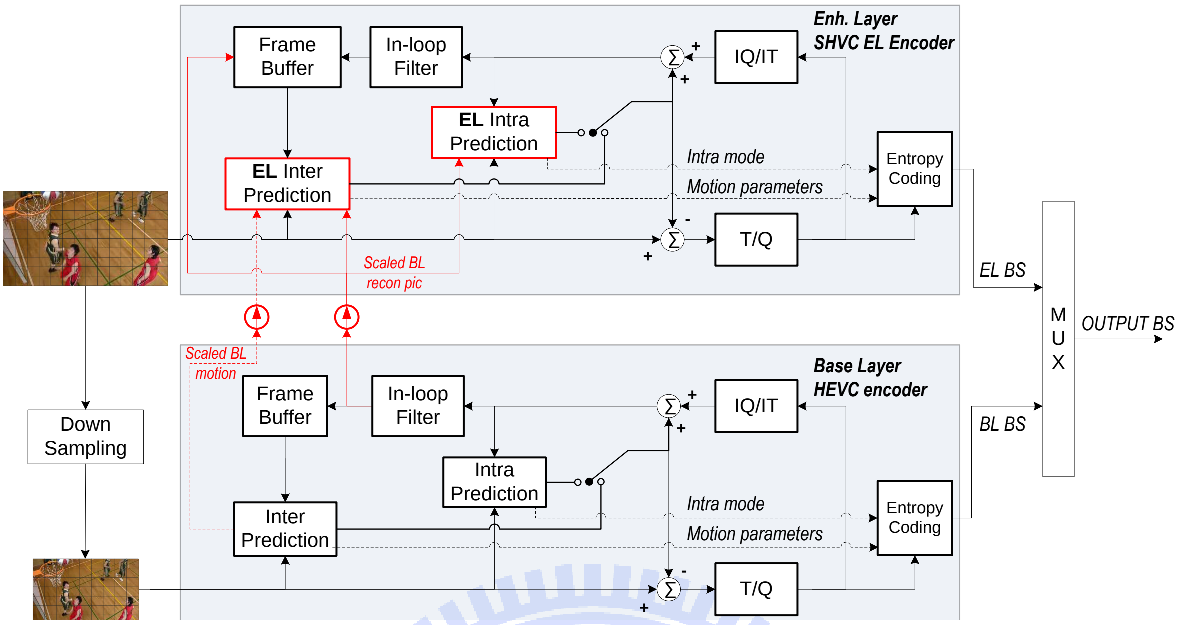


Figure 2: A two-layer spatial scalability encoder using HEVC encoding modules

In [1], two signaling mechanisms, namely the RefIdx and TextureRL approaches, are proposed for ILP. Those tools are described as follows.

ILP in RefIdx-based SHVC: The BL reconstructed picture (which is referred to as the inter-layer reference picture) and the temporal reference picture are included in the reference picture lists, so that the inter-layer motion or texture prediction can be achieved by simply using the existing inter-frame prediction mechanisms. This approach has benefit of being able to reuse most of the single layer HEVC design, except few high-level (e.g. slice level) syntax changes, for encoding and decoding the enhancement layer.

ILP in TextureRL-based SHVC: This approach allows to be changed the low-level (e.g. CU- and/or PU-level) syntax and decoding process. Specifically, the inter-layer motion and the texture prediction are achieved as follows

- Inter-layer motion prediction: The *merge mode* in HEVC is to be modified. In particular, the motion parameters of the collocated BL block is also used to form the merge candidate list. This merge candidate is derived at the location collocated to the central position of the current PU and to be considered as the first candidate in the merge list.
- Texture prediction: This mechanism is achieved by introducing a new prediction mode, named as Intra-BL mode. This mode has to compete with other intra-frame and inter-frame prediction mode in the Rate-Distortion Optimization (RDO) process to determine the best prediction mode in terms of lowest rate-distortion (RD) cost.

Without being constrained to use inter-frame prediction mechanisms, TextureRL approach is more flexible in constructing ILP. For example, a new prediction mode may be created at the PU level to better utilize both the BL and EL information for prediction. The price paid for achieving this is however a less compatible codec for the EL. Furthermore, even though the TextureRL approach has more flexibility, the RefIdx approach can already achieve most of the gain provided by the TextureRL [13][14], making the it a less favorable approach. But, many believe the flexibility of the TextureRL has not been employed to the best advantage. As a result, in SHVC, multiple designs related to combined prediction are proposed in order to explore the potential of the TextureRL approach in forming a better ILP mechanism. Those proposals use the BL and EL information to improve the coding efficiency of the EL. Specially, the intra prediction improvements based on the reconstructed BL will be described in the following sections.

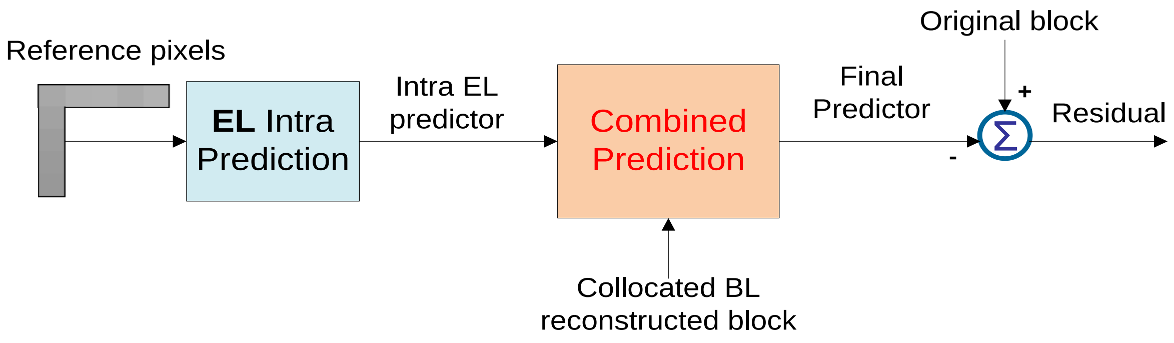


Figure 3: Combined intra prediction at the enhancement layer in SHVC

2.3. Combined Intra Prediction

Conceptually, in intra prediction, it is difficult to predict the whole block accurately [4] and usually, the bottom right pixels in the block are predicted less accurately because they are far away from the reference and the correlation between them is weaker. Therefore, this inaccurate prediction results in large residual information that reduces the coding efficiency. The inaccuracy in prediction becomes worse when the prediction block size increases. This issue needs to be taken into account especially in the new standard when the coding unit size increases from 16x16 in H.264/AVC to 64x64 in HEVC. As a result, it raises a need to have a mechanism to compensate for this lack of prediction by using the extracted information from the BL.

Figure 3 represents the combined process introduced in SHVC. The '*Combined Prediction'* block is generally used to extract the information from the BL to compensate for the EL intra predictor, resulting in the difference between the final prediction and the input block being minimized. Consequently, the prior works in the context of intra prediction improvement at the EL will be presented as follows.

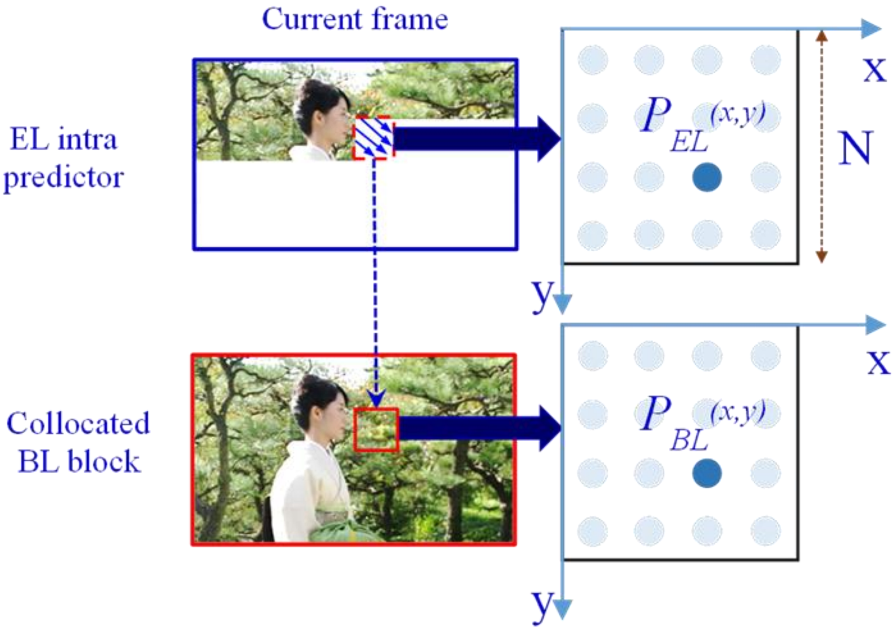


Figure 4: Intra DC Correction algorithm, $P_{EL}(x,y)$ and $P_{BL}(x,y)$ are the pixel values of the EL intra predictor and collocated BL reconstructed block, N is the width (or height) of the current block.

2.3.1. Intra DC Correction

The intra DC correction (IDCC) in [7] is based on the observation that the mean value (DC value) of the EL prediction block can sometimes be better estimated from the collocated BL block. As a result, the DC value of the EL intra predictor is replaced by that of the collocated BL block. The DC value of the EL intra predictor and its collocated BL block is calculated by the mean value of all pixels located in the block as follows

$$DC_{EL} = \frac{\sum P_{EL}(x,y)}{N^2} \tag{1}$$

$$DC_{BL} = \frac{\sum P_{BL}(x,y)}{N^2} \tag{2}$$

The final prediction is derived, at the pixel level, as follow

$$P_{Final}(x,y) = P_{EL}(x,y) - DC_{EL} + DC_{BL} \tag{3}$$

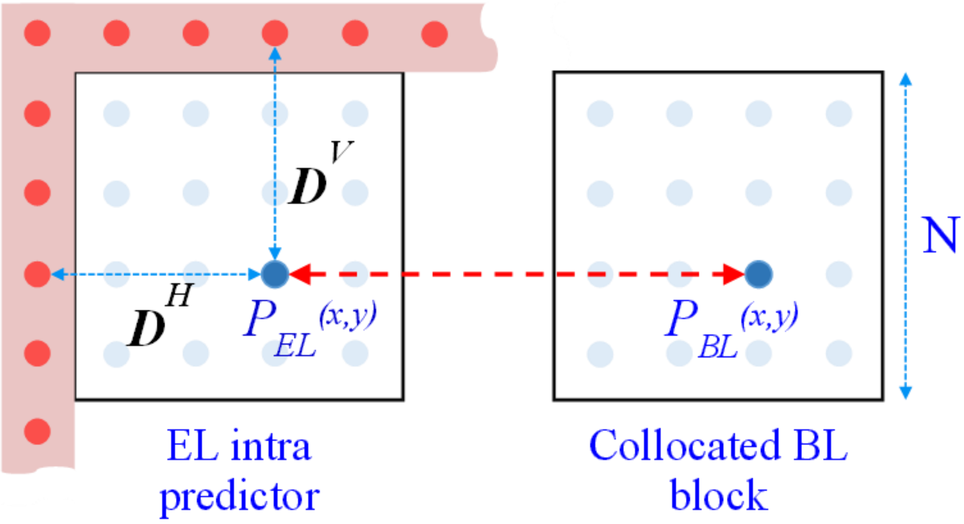


Figure 5: Weighted Intra Prediction algorithm, *P_{EL}(x,y)* and *P_{BL}(x,y)* are the pixel values of the EL intra predictor and collocated BL pixel, D^H and D^V are the horizontal and vertical distances of current pixel to its references

2.3.2. Weighted Intra Prediction

The algorithm proposed in [6] is also an effort to improve the intra prediction in the EL by employing the reconstructed BL texture. In general, as the distance between the pixel position and its reference sample increases, the spatial correlation decreases so that the accuracy of the prediction value is reduced. In order to compensate for this, a weighting method is used, in which as the distance between the $P_{EL}(x,y)$ and its references increases, the weight value for the $P_{BL}(x,y)$ also increases. Therefore, the reduction of accuracy of $P_{EL}(x,y)$ due to the increasing distance can be compensated for by exploiting the corresponding value from BL. Finally, in order to generate the final prediction signal, the following processes are applied

$$P^V(x,y) = (N - D^H) \times P_{EL}(x,y) + D^H \times P_{BL}(x,y) \tag{4}$$

$$P^H(x,y) = (N - D^V) \times P_{EL}(x,y) + D^V \times P_{BL}(x,y) \tag{5}$$

$$P_{Final}(x,y) = (P^V(x,y) + P^H(x,y) + N) >> (log_2(N) + 1) \tag{6}$$

CHAPTER 3

Mode-Dependent Pixel-Based Weighted Intra Prediction (MPWIP)

This chapter presents the proposed mode-dependent pixel-based weighted intra prediction. We start by introducing its concept of operations, showing first the decomposition of the EL intra predictor and the BL reconstructed block into their respective AC and DC components and then the combination of these components based on a pixel-based weighting scheme. As we shall see, the weight value to associate with each component is designed to be a function of the prediction pixel's position within the block, which we refer hereafter to as the *weighting function* (it represents the weighting scheme from the perspective of a single component [e.g. the AC or the DC component at the BL or at the EL], characterizing the contribution of one component to estimating a set of sample intensities), and is optimized based on a least-squares fit to a set of training data. Lastly, a thorough analysis is conducted on

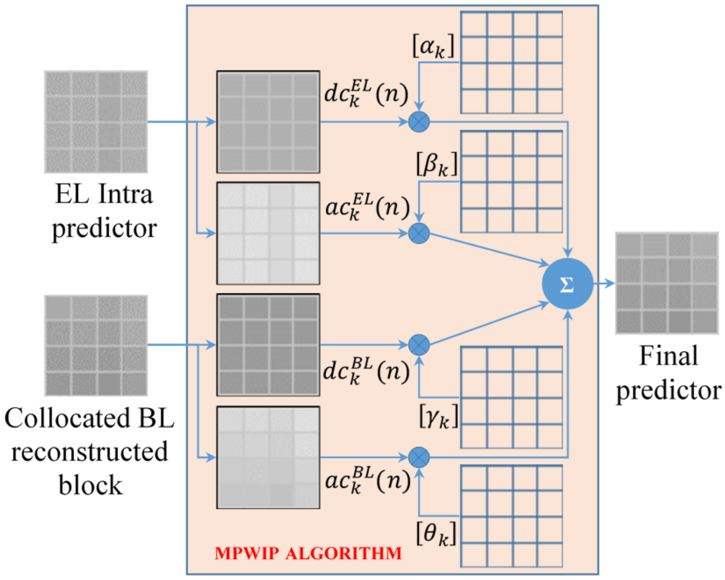


Figure 6: Proposed Pixel-based Weighted Intra Prediction scheme

how coding parameters, such as the EL intra prediction mode, QP setting, and prediction block size, affect the weighting functions of different components.

3.1. Pixel-based Weighted Intra Prediction

3.1.1. Concept of Operations

The ideas of the proposed scheme are essentially a combination of the two prior works [6] and [7]. As depicted in Figure 6, the texture information of the EL intra predictor and BL reconstructed block are first decomposed into the DC and AC components, where the DC components are computed as a prediction block with all its pixels taking the average value of the input block, i.e., the EL intra predictor or the BL reconstructed block, and the AC components are formed from the residual signals produced by subtracting the DC components from their respective inputs. Each of these components is then weighted by a separate pixel-based weighting scheme and the results are summed together to form the final predictor.

3.1.2. Least-Squares Solution

Obviously, how different components are weighted has a crucial effect on the resulting prediction performance. We wish to find a set of weighting functions so that the resulting prediction residual can be minimized. This problem can be solved using the well-known Least-Squares (LS) method.

To ease the understanding of the subsequent discussion, we adopt the following notations: **bold lower-case** letters represent vectors, **BOLD UPPER-CASE** letters denote matrices, and *italicized lower-case* letters are scalars. Moreover, we use $\mathbf{a}_k = [a_k(1)\ a_k(2)\ \dots\ a_k(n)]^{\mathrm{T}}$ and $\mathbf{b}_k = [b_k(1)\ b_k(2)\ \dots\ b_k(n)]^{\mathrm{T}}$ to represent the predictor values at pixel k that are extracted from the EL intra predictors and the BL reconstructed blocks, respectively, in n collected data from the training process. Similarly $\mathbf{ac}_k = [ac_k(1)\ ac_k(2)\ \dots\ ac_k(n)]^{\mathrm{T}}$ and $\mathbf{dc}_k = [dc_k(1)\ dc_k(2)\ \dots\ dc_k(n)]^{\mathrm{T}}$ denote respectively the corresponding values from the AC and DC components. Thus, we have

$$\mathbf{a}_k = \underbrace{(\mathbf{a}_k - \mathbf{dc}_k^{EL})}_{\mathbf{ac}_k^{EL}} + \mathbf{dc}_k^{EL} \tag{7}$$

$$\mathbf{b}_k = \underbrace{(\mathbf{b}_k - \mathbf{dc}_k^{BL})}_{\mathbf{ac}_k^{BL}} + \mathbf{dc}_k^{BL} \tag{8}$$

With these, the final prediction signal $\mathbf{p}_k = [p_k(1)\ p_k(2)\ \dots\ p_k(n)]^{\mathrm{T}}$ at pixel k can be written as:

$$\mathbf{p}_k = \mathbf{X}_k . \mathbf{w}_k \tag{9}$$

where $\mathbf{X}_k = [\mathbf{ac}_k^{EL}\ \ \mathbf{dc}_k^{EL}\ \ \mathbf{ac}_k^{EL}\ \ \mathbf{dc}_k^{BL}]$ and $\mathbf{w}_k = \left[\alpha_k\ \ \beta_k\ \ \gamma_k\ \ \theta_k\right]^{\mathrm{T}}$ is a weight vector whose elements are the weight values to associate with the four components for prediction at pixel k. Specifically, the weight vector represents the weighting scheme from the perspective of a single pixel, describing how the corresponding samples from different components contribute to estimating a current pixel’s intensity.

With reference to the notations above, we further denote by $\mathbf{o}_k = [o_k(1)\ o_k(2)\ \ldots\ o_k(n)]^{\mathrm{T}}$ the target pixels at position k in n collected blocks, whose intensity values are to be estimated. The problem of determining the optimal weight vector $\mathbf{w}^*{}_k$ in the least-squares-error sense can then be formulated as follows:

$$\mathbf{w}^*{}_k = \underset{w_k}{\operatorname{argmin}}(\mathbf{o}_k - \mathbf{X}_k.\mathbf{w}_k)^2 \qquad (10)$$

From the Linear Algebra theorem, it has the closed-form solution

$$\mathbf{w}^*{}_k = (\mathbf{X}_k^{\mathrm{T}}.\mathbf{X}_k)^{-\mathbf{1}}.\mathbf{X}_k^{\mathrm{T}}.\mathbf{o}_k \qquad (11)$$

By varying the index k and repeating the same process, we can obtain the weight vectors for different pixel positions and thus the weighting functions for all four components.

3.1.3. Training Process

In order to collect the data for the basis functions to compute the optimal weighting function, the training process is introduced. First, to ensure the collected data is most appropriate, the proposed algorithm will be applied to produce a new prediction mode and this mode has to compete with all conventional modes in the rate-distortion optimization (RDO) process at the EL to find the best mode with lowest rate distortion (RD) cost. Then, those blocks coded in the proposed algorithm will be used to compute the optimal weighting functions with respect to Eq. (11).

It is observed that according to the Eq. (10), on each iteration of training process, the initial values of the weighting functions needs to be assigned. Specifically, in our training process the weight value corresponds to the average value of the EL and the

BL's texture information which is utilized for the first iteration. Then, the updated weighting functions are referred to as the input weighting functions for the next iteration; the process is repeated for all the sequences in the training set. Finally, the criteria to terminate the training process are determined with respect to a general consensus is that if the weighting functions are optimized, the mean square error (MSE) of all the coded pixels in the current iteration should be the relatively smaller than that of other iterations. In particular, the training process is to be terminated according to two criteria: 1) The weighting functions are stable (that is, they do not vary considerably compared to previous iteration) and 2) The absolute difference in MSE value between the current iteration and its successive previous iteration is below 1% of MSE value of the successive previous iteration.

However, the resulting weighting functions are only optimized for a specific iteration of the training process. In addition, the sequences of the training set differ from the sequences of the test set (because it is important to test a model by data which is different from that used to develop it); therefore, the obtained weighting functions used to find the bit rate savings for the test sequences would be referred to as *the weighting functions that are resulted from the training process*.

3.2. Weighting Function

To gain a better understanding of how different components should be weighted in forming a better predictor, this section provides an in-depth analysis of the weighting functions with different components against 1) prediction mode taken by the EL, 2) QP setting of BL and EL, and 3) prediction block size.

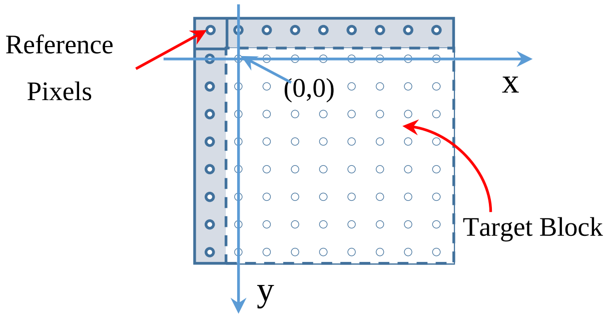


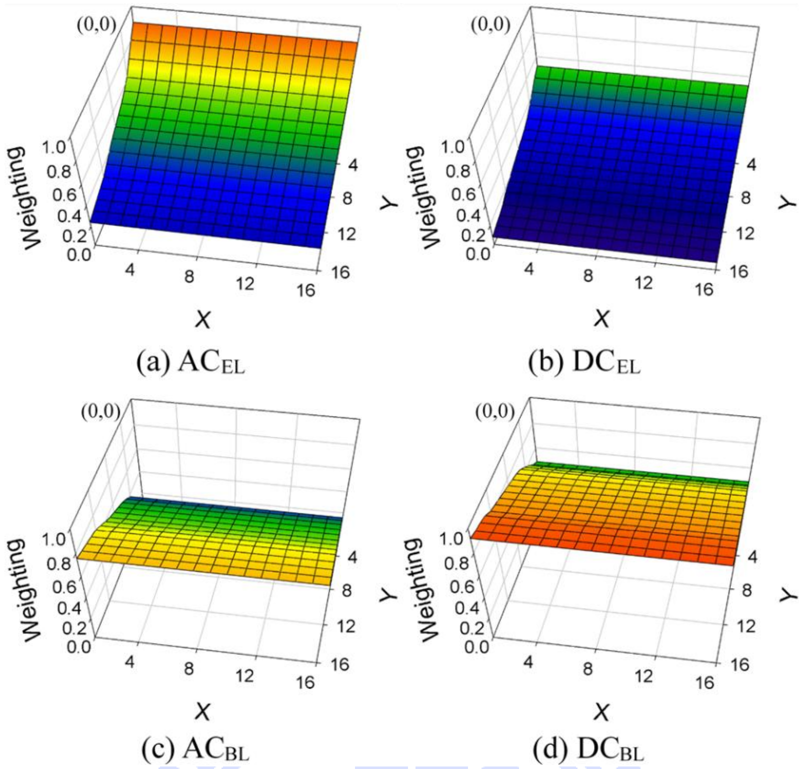
Figure 7: Pixel coordinate system showing the pixel at (0,0)

For notation, the QP value of BL and EL is specified by a two-tuple representation QP(QP_{BL}, QP_{EL}), and the coordinate system shown in Figure 7 is used throughout the discussion that follows.

3.2.1. Effect of Intra Prediction Mode

This section investigates the effect of the intra prediction mode on the weighting function. Here the prediction mode refers to the intra prediction direction used to generate the EL predictor. Currently, our weighted scheme is restricted to the cases where the EL predictor is produced with Horizontal, Vertical, DC or Planar mode.

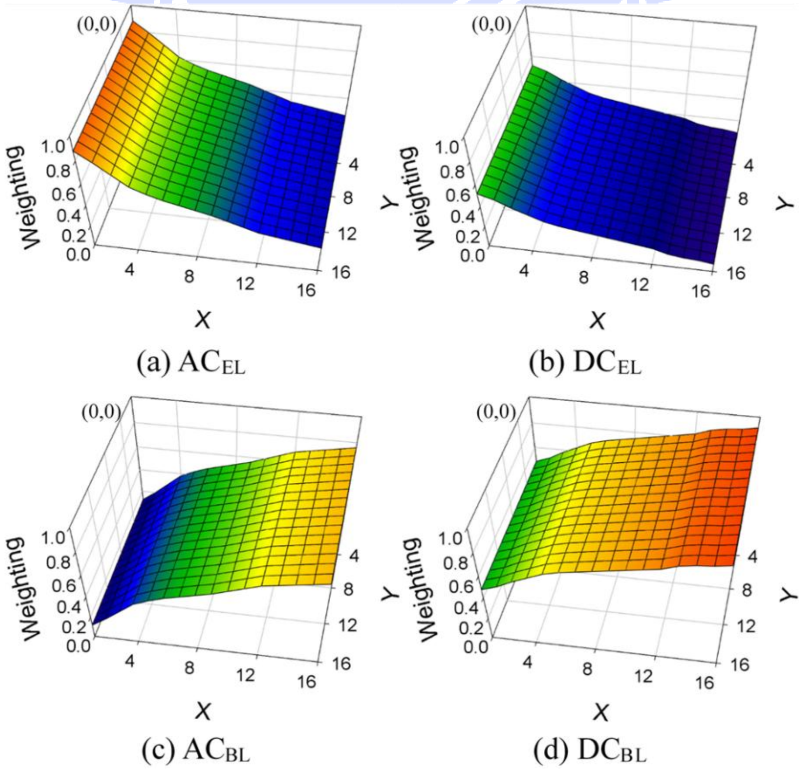
Figure 8 show the weighting functions for Vertical mode. It can be seen that those associated with the components from the same layer have a similar waveform, although their magnitudes differ considerably. Moreover, the weight value for the DC component of the EL is seen to be mostly lower than that of the BL, which justifies the IDCC algorithm's substitution of the BL's DC value for the EL's. By comparing, Figure 9, Figure 10, and Figure 11, we can further observe that the weighting functions vary with the prediction mode with which the EL predictor is formed—i.e., they are mode dependent.



(a) AC_{EL} (b) DC_{EL}

(c) AC_{BL} (d) DC_{BL}

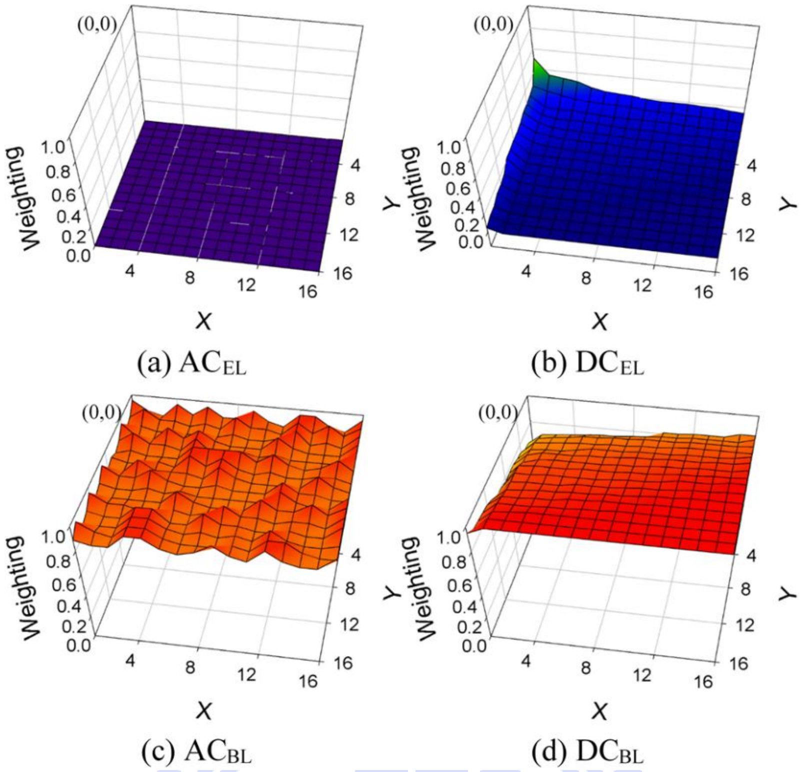
Figure 8: Vertical mode, block size of 16x16, QP(30,30). Each figure corresponds to the weighting function of (a) AC_{EL}, (b) DC_{EL}, (c) AC_{BL}, (d) DC_{BL}



(a) AC_{EL} (b) DC_{EL}

(c) AC_{BL} (d) DC_{BL}

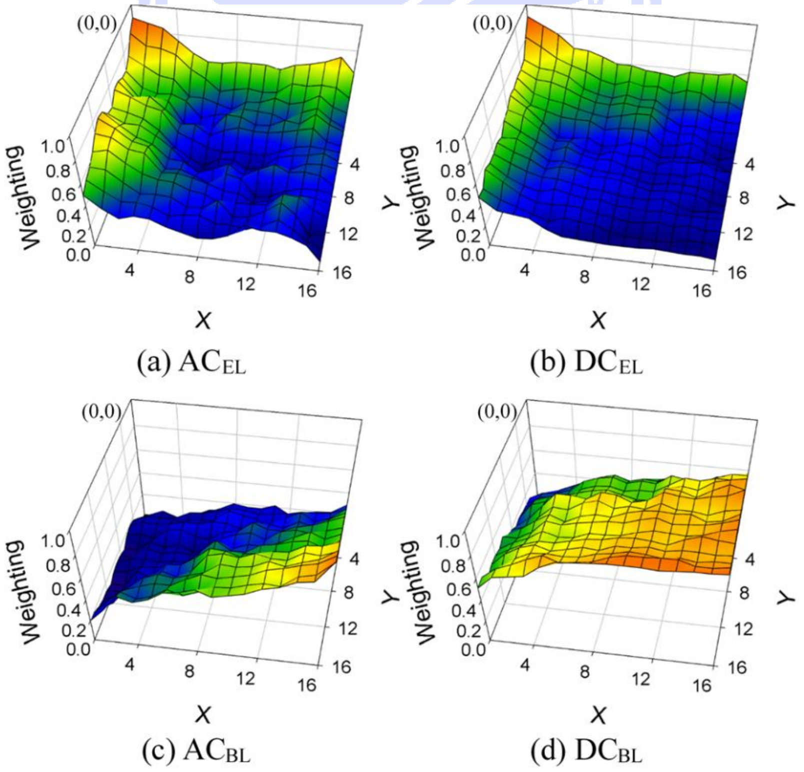
Figure 9: Horizontal mode, block size of 16x16, QP(30,30). Each figure corresponds to the weighting function of (a) AC_{EL}, (b) DC_{EL}, (c) AC_{BL}, (d) DC_{BL}



(a) AC_{EL} (b) DC_{EL}

(c) AC_{BL} (d) DC_{BL}

Figure 10: DC mode, block size of 16x16, QP(30,30). Each figure corresponds to the weighting function of (a) AC_{EL}, (b) DC_{EL}, (c) AC_{BL}, (d) DC_{BL}. The AC_{EL} component of the DC mode is not available and it is not weighted in the experiments.



(a) AC_{EL} (b) DC_{EL}

(c) AC_{BL} (d) DC_{BL}

Figure 11: Planar mode, block size of 16x16, QP(30,30). Each figure corresponds to the weighting function of (a) AC_{EL}, (b) DC_{EL}, (c) AC_{BL}, (d) DC_{BL}.

Another interesting point to be noted in Figure 10 is that, when the EL is coded in DC mode, in which case the EL predictor contains no AC component, the BL's texture information dominates the creation of the final predictor, but the contribution from the EL is not insignificant.

Finally shown in Figure 11 are the weighting functions that resulted from the Planar mode. We see that the EL predictor is weighted more heavily in forming a prediction of pixels in the top-left corner while the BL counterpart contributes more to the rest of pixels, especially those sitting in the bottom-right quadrant. This can be explained by the way that the EL predictor is constructed. Specifically, the Planar mode in HEVC produces the EL predictor using bilinear interpolation, with the reconstructed pixels at block boundaries serving as reference. While pixels on the top and to the left of a current block have been reconstructed previously, those at the bottom or to the right are yet to be determined. In the current implementation, their sample values are estimated by replicating that of the top-right and bottom-left pixels, respectively. The fact that the value of these pixels are weakly correlated with that of pixels in the bottom-right quadrant accounts for why the BL is assigned with a higher weight value there.

3.2.2. Effect of QP Setting

This section studies the effect of the QP setting (QP values assigned to the BL and the EL) on the weighting functions. Plotted in Figure 12 is the weighting function of the EL's DC component along the slice of Y=10 for various QP settings (with the QP_{BL} ranging from 22 to 34, increasing by 4, and delta QP=0 or 2, delta QP is the difference in QP values between the QP_{EL} and the QP_{BL}); the results shown correspond to the

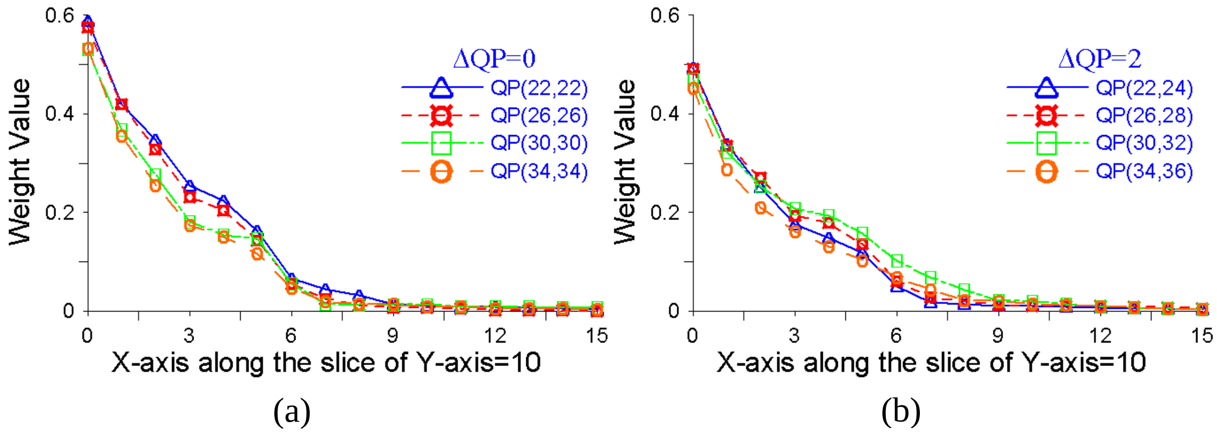


Figure 12: The curves of DC component in the EL of Horizontal mode on different QP settings in two sets of coding configurations. (a) ΔQP=0: QP_{EL} = QP_{BL}, (b) ΔQP=2: QP_{EL} = QP_{BL} +2

Horizontal mode. It is noted that the observations made in this case also applies to the other cases.

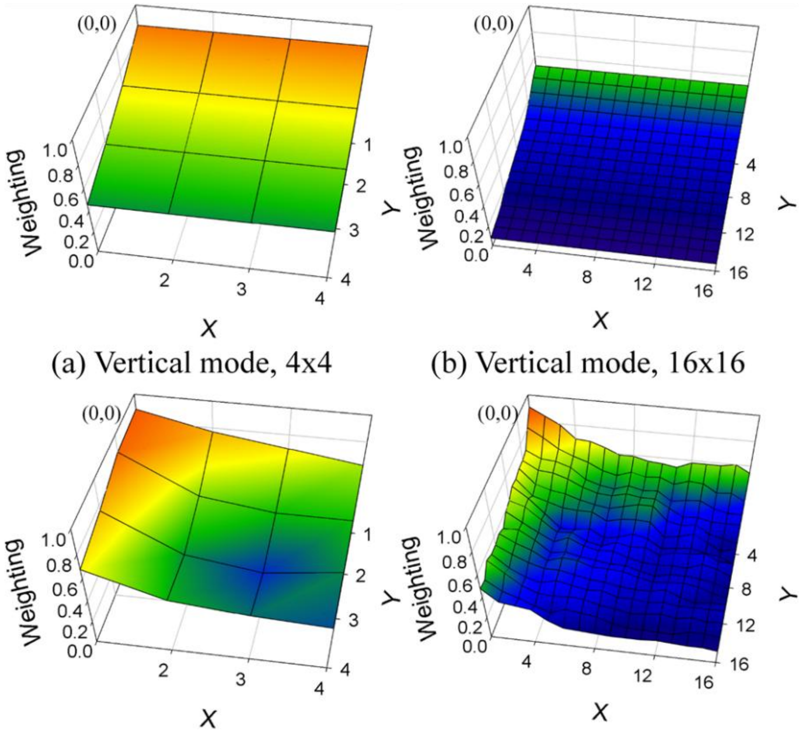
As expected, all these weighting functions decrease with the increasing X value because of the nature of Horizontal prediction and this is a result that we have seen before. Of more interest is the observation that the EL tends to be weighted more heavily when the EL is coded at better quality using a smaller QP value (note that given the same QP_{BL}, a smaller delta QP leads to a smaller QP_{EL} and vice versa) or the BL is coarsely quantized with a larger QP. This agrees with the general observation that when coded at better quality, the EL reference pixels correlate more strongly with the prediction pixels. The same argument is equally applicable to the case having a poorly coded BL. Somewhat to our surprise; the extent to which these weighting functions differ with each other is not that significant.

Another interesting observation from the weight values of the QP settings in the same set is that even though the difference between the smallest and the largest QP settings are considerable (with the value of 22 for the first and 34 for the latter case), the variance in the magnitude of those waveforms is insignificant.

The experimental results that showed the effect of the QP setting on weighting functions in terms of bit rate savings are provided in details in Chapter 4. With respect to these experimental results, the effect of the QP setting on the weighting functions in terms of bit rate savings is insignificant. Specifically, the experimental results showed that 1) the weighting functions for different QP settings within the same delta QP group can be unified by sharing a set of weighting functions of any QP setting; 2) the weighting functions for different QP settings between two delta QP groups can be unified by a constraint that the QP settings having the same QP value for the BL can share the same set of weighting functions. Therefore, these observations can lead to an opportunity to unify the weighting functions for the QP settings in the same delta QP set and/or all the QP settings in the common test conditions.

3.2.3. Effect of Prediction Block Size

This last investigation shows how the prediction block size affects the weighting functions. The effect of the prediction block size is somewhat expectable (the higher weight value for the EL’s components will be given to the smaller block size); however, the texture information in the BL perhaps varies the expectation. Therefore, we need to consider the interaction of the BL texture information. As a result, the analysis on the effect of the prediction block size is necessary.



(a) Vertical mode, 4x4

(b) Vertical mode, 16x16

(c) Planar mode, 4x4

(d) Planar mode, 16x16

Figure 13: Waveforms of weighting functions for DC component in the EL of Vertical and Planar mode at different block size levels.

To this end, Figure 13 contrasts the weighting function of the EL's DC component for 4x4 and 16x16 block sizes. Results are given for Vertical and Planar modes. As expected, the simulation results show that the EL predictor tends to have a higher weight value across the entire prediction block when the block size is smaller. This is understandable given that directional intra prediction usually performs more efficiently with a smaller block size and that the EL reference pixels are subject to less coding error.

3.2.4. Summary

We can summarize our findings so far as follows:

- The weighting functions depend on the direction of the prediction mode taken by the EL intra predictor, i.e. they are mode dependent.
- The DC and AC components from the same layer have sets of weighting functions that are similar in waveform but differ in magnitude, i.e. the separation of DC and AC components seems beneficial.
- The weighting functions of Horizontal and Vertical modes are related to each other mainly by a transpose operation.
- The effect of the QP setting on the weighting functions is insignificant in terms of bit rate savings in the common test conditions.
- The EL tends to be weighted more heavily in the case of smaller block sizes.

CHAPTER 4

Experimental Results

4.1. Test Conditions

Experimental results provided in this chapter are produced following mainly the All Intra (AI) common test conditions specified in [8]. In particular, only the results for mandatory tests, in which the base layer is HEVC coded, are presented. Moreover, for the 2-layer spatial scalability, the ratio of the EL's resolution to that of BL is limited to 2x and 1.5x, although in principle any resolution ratios between the layers, including the ratio of 1, can be considered. Table 1 details the test sequences used. It is noteworthy that the weight values used in our pixel-based weighted intra prediction scheme are obtained based on a separate set of training sequences, as given in Table 2. Generally, it is important to test a model against data which is outside of the samples used to develop it.

Table 1: Test set of video sequences

Class	Sequence name	Frame count	Frame rate	Bit depth	BL Resolution	EL Resolution	All-Intra Configurations
A	Traffic	150	30fps	8	1280x800	2560x1600	Spatial 2x
A	PeopleOnStreet	150	30fps	8	1280x800	2560x1600	Spatial 2x
B	Kimono	240	24fps	8	960x540 1280x720	1920x1080 1920x1080	Spatial 2x Spatial 1.5x
B	ParkScene	240	24fps	8	960x540 1280x720	1920x1080 1920x1080	Spatial 2x Spatial 1.5x
B	Cactus	500	50fps	8	960x540 1280x720	1920x1080 1920x1080	Spatial 2x Spatial 1.5x
B	BasketballDrive	500	50fps	8	960x540 1280x720	1920x1080 1920x1080	Spatial 2x Spatial 1.5x
B	BQTerrace	600	60fps	8	960x540 1280x720	1920x1080 1920x1080	Spatial 2x Spatial 1.5x

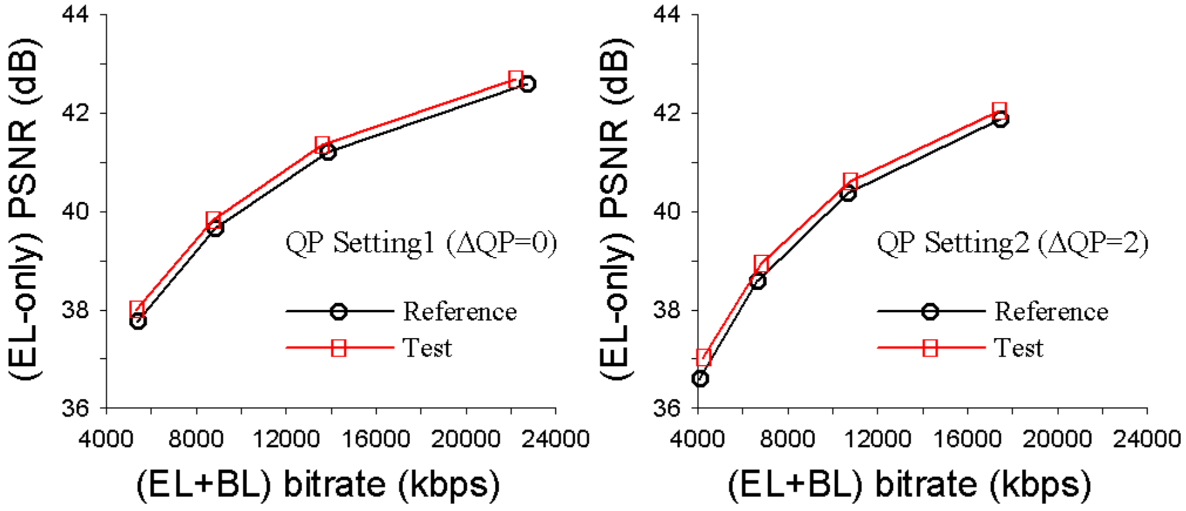
Table 2: Training set of video sequences

Class	Sequence name	Frame count	Frame rate	Bit depth	BL Resolution	EL Resolution	All-Intra Configurations
A	Nebuta	300	60fps	10	1280x800	2560x1600	Spatial 2x
A	SteamLocomotive	300	60fps	10	1280x800	2560x1600	Spatial 2x
B	Blue Sky	217	25fps	8	960x540 1280x720	1920x1080 1920x1080	Spatial 2x Spatial 1.5x
B	Tennis	240	24fps	8	960x540 1280x720	1920x1080 1920x1080	Spatial 2x Spatial 1.5x
B	Riverbed	250	25	8	960x540 1280x720	1920x1080 1920x1080	Spatial 2x Spatial 1.5x
B	Troy	300	24fps	8	960x540 1280x720	1920x1080 1920x1080	Spatial 2x Spatial 1.5x
B	Station	313	25fps	8	960x540 1280x720	1920x1080 1920x1080	Spatial 2x Spatial 1.5x
B	Pedestrian Area	375	24fps	8	960x540 1280x720	1920x1080 1920x1080	Spatial 2x Spatial 1.5x
B	Rush Hour	500	25fps	8	960x540 1280x720	1920x1080 1920x1080	Spatial 2x Spatial 1.5x
B	Tractor	690	25fps	8	960x540 1280x720	1920x1080 1920x1080	Spatial 2x Spatial 1.5x

Table 3: Test set of quantization parameter values

Scalability	QP of BL	ΔQP = QP of EL – QP of BL
Spatial 2x	22, 26, 30, 34	0, 2
Spatial 1.5x	22, 26, 30, 34	0, 2

Table 3 defines the quantization parameter values used for the I-frames in the base and enhancement layers of a sequence for the HEVC base layer case.



(a) ΔQP=0 (b) ΔQP=2

Figure 14: An example of rate-distortion curves for two sets of encoding configurations. Four QP values of BL in each set are 22, 26, 30, and 34.

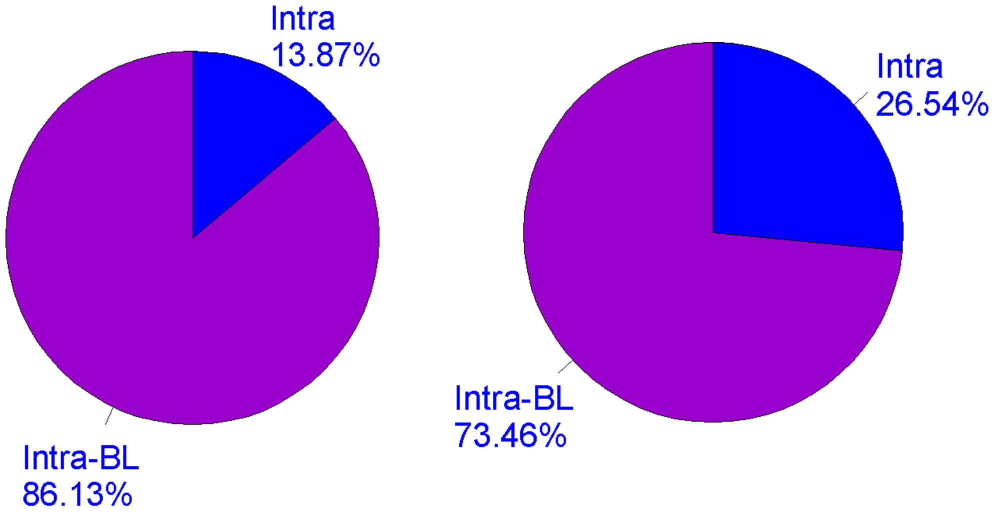
4.2. Coding Performance

The coding performance of the proposed scheme is quantified by measuring the BD-rate savings [2] relative to the SHM 1.0 anchor. In particular, the results presented in Table 4 correspond to the average numbers taken over two sets of encoding configuration, as suggested by the common test conditions [8]. The method used to calculate the values reported in all performance tables will be described as follows. First, the BD-rate calculation for the single layer coding will be used to calculate the BD-rate savings in each set of QP configurations (ΔQP=0, ΔQP=2) as depicted in Figure 14. The average numbers presented for every coding performance savings table are the average values of these BD-rate savings obtained from the aforementioned two sets.

From Table 4, the overall Y-BD-rate gain over the anchor is 1.0% for the AI-2x case and 0.5% for the AI-1.5x case. The coding gain achieved, however, is highly variable over the test sequences. As an example, the smallest gain is in the

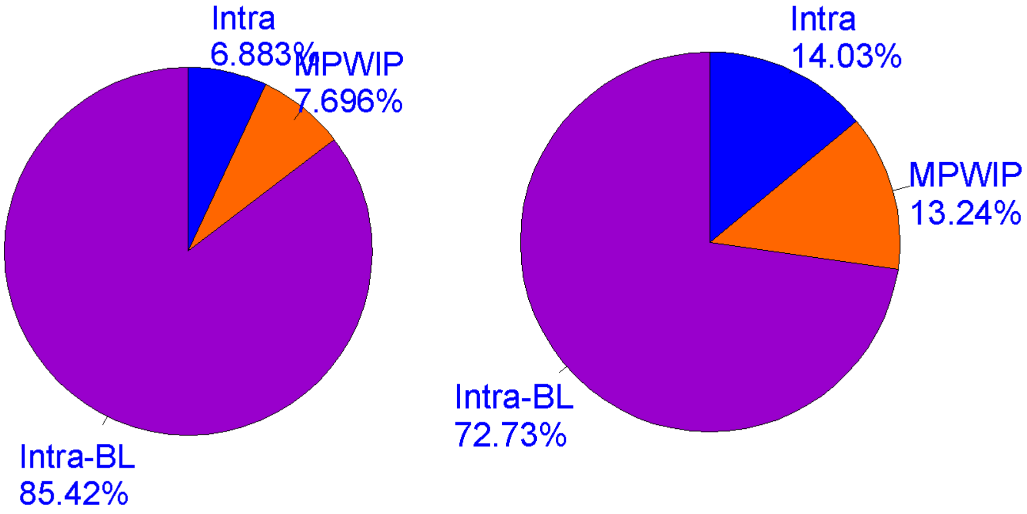
Table 4: Performance of the MPWIP with respect to SHM 1.0

		AI HEVC 2x			**AI HEVC 1.5x**		
		Y	U	V	Y	U	V
Class A	Traffic	-0.7%	-0.6%	-0.6%	N/A		
	PeopleOnStreet	-0.6%	-0.7%	-0.4%			
Class B	Kimono	-0.4%	-0.4%	-0.5%	-0.2%	-0.3%	-0.3%
	ParkScene	-0.3%	-0.3%	-0.4%	-0.1%	-0.2%	-0.3%
	Cactus	-1.0%	-1.2%	-1.3%	-0.3%	-0.6%	-0.6%
	BasketballDrive	-2.4%	-2.7%	-2.2%	-1.5%	-1.6%	-1.3%
	BQTerrace	-1.3%	-1.6%	-1.8%	-0.4%	-0.6%	-0.7%
Overall (EL+BL)		-1.0%	-1.1%	-1.0%	-0.5%	-0.7%	-0.6%



(a) ParkScene (b) BasketballDrive

Figure 15: The statistical mode distribution of two test sequences at the enhancement layer in SHM-1.0: (a) The *'ParkScene'* sequence, (b) The *'BasketballDrive'* sequence



(a) ParkScene (b) BasketballDrive

Figure 16: The statistical mode distribution of two test sequences at the enhancement layer in the proposed design: (a) The *'ParkScene'* sequence, (b) The *'BasketballDrive'* sequence

‘ParkScene’ sequence, with only a 0.3% and 0.1% gain for AI-2x and AI-1.5x, respectively, whereas a much higher improvement is observed in the *‘BasketballDrive’* sequence, reaching up to 2.4% and 1.5%, respectively. The observation above can be explained by the statistical study on mode distribution and the characteristics of each sequence.

Conventionally, in the AI configuration of the common test conditions, at the EL, there are two prediction modes; Intra and Intra-BL. In the proposed design, the proposed algorithm is applied to create a new prediction mode, called MPWIP; this prediction mode has to compete with other conventional modes in the rate distortion optimization (RDO) process to find the best mode with the lowest rate distortion (RD) cost. Therefore, generally speaking, the more pixels coded in this new mode the higher the bit rate savings are expected to be achieved.

The statistical results will be analyzed in this section. Figure 15 shows the mode distribution diagram of the reference software (SHM-1.0) which includes only two conventional prediction modes. It can be seen that the percentage of intra-coded pixels in the *‘BasketballDrive’* sequence is considerably higher than that in the *‘ParkScene’* sequence. Specifically, it is observed that 26.54% more pixels are intra-coded in the *‘BasketballDrive’* sequence, while half as much of that percentage (i.e. 13.87%) is found in the *‘ParkScene’* sequence. This can be explained by the characteristics of each sequence. The *‘ParkScene’* sequence is a highly-textured sequence while the *‘BasketballDrive’* sequence contains more homogeneous regions.

Figure 16 shows the mode distribution diagram of the proposed design. The results show that the percentage of pixels coded in the new mode of the *‘ParkScene’*

Table 5: Performance of the IDCC with respect to SHM 1.0

		AI HEVC 2x			**AI HEVC 1.5x**		
		Y	U	V	Y	U	V
Class A	Traffic	-0.1%	0.0%	0.0%	N/A		
	PeopleOnStreet	-0.1%	-0.1%	0.0%			
Class B	Kimono	-0.1%	0.0%	0.0%	0.0%	0.1%	0.0%
	ParkScene	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	Cactus	-0.2%	-0.2%	-0.3%	0.0%	-0.1%	-0.1%
	BasketballDrive	-0.5%	-0.3%	-0.4%	-0.1%	-0.1%	-0.1%
	BQTerrace	-0.2%	-0.2%	-0.2%	-0.1%	-0.1%	-0.1%
Overall (EL+BL)		-0.2%	-0.1%	-0.1%	-0.1%	0.0%	-0.1%

Table 6: Performance of the WIP with respect to SHM 1.0

		AI HEVC 2x			**AI HEVC 1.5x**		
		Y	U	V	Y	U	V
Class A	Traffic	-0.2%	-0.2%	-0.2%	N/A		
	PeopleOnStreet	-0.1%	-0.4%	-0.2%			
Class B	Kimono	0.0%	-0.1%	-0.1%	0.0%	0.0%	0.0%
	ParkScene	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%
	Cactus	-0.3%	-0.3%	-0.6%	-0.1%	-0.2%	-0.4%
	BasketballDrive	-0.7%	-1.8%	-1.7%	-0.3%	-0.7%	-0.7%
	BQTerrace	-0.7%	-1.1%	-1.6%	-0.2%	-0.5%	-0.7%
Overall (EL+BL)		-0.3%	-0.6%	-0.6%	-0.1%	-0.3%	-0.4%

sequence is only 7.696%, while 13.24% of that was found in the *'BasketballDrive'* sequence. As a result, the higher bit rate savings can be achieved in the latter sequence. In particular, for both test sequences, most of the pixels coded in the new mode resulted from those that were intra-coded or, alternatively, the proposed algorithm mostly improved the intra-frame prediction. Therefore, it can be concluded that the proposed algorithm works best in the sequences that contain more homogeneous regions.

For comparison with the IDCC and WIP algorithms, their respective BD-rate savings relative to the anchor are shown in Table 5 and Table 6. It can be seen that both schemes offer a much smaller BD-rate savings than our MPWIP. Specially, that of the IDCC ranges from 0.0% to 0.5% in all test cases, with an overall savings of no more than 0.2%. The WIP, although performing relatively better in some sequences (such as *‘BasketballDrive’* and *‘BQTerrace’*), shows a similar performance to the IDCC in terms of Y-BD-rate.

4.3. Simplification

The superior coding performance of our MPWIP comes at the cost of additional memory requirements in both the encoder and decoder for storing weight tables. In our current implementation, the luminance and chrominance components use separate weight tables for different block sizes (7 block sizes in total). Moreover, according to our analysis in Chapter 3, the weight tables for Horizontal, Vertical, Planar and DC modes are distinct from each other. The situation is further complicated by the QP setting, as we found the weighting functions also vary with the QP setting of the BL and the EL. As a result, a total of 840 weight tables are needed, which calls for a memory space of 795 kilobytes (with an assumption that a single-precision floating-point format occupies 32 bits [4 bytes]). Obviously, 795kB of required memory storage is a significant cost for an on-chip memory design.

4.3.1. The Unification to MPWIP

In an attempt to reduce the number of weight tables, we began to analyze the effect of the QP settings on the weighting functions in terms of bit rate savings; two experiments were carried out.

The first experiment is to examine the effect of the QP setting within the delta QP group and the second one is to examine the effect between two QP setting groups. Specifically, in the former case, we first took out a set of weighting functions obtained

Table 7: Comparison performance of Y-BD-rate savings of an experiment in Test Group A vs. MPWIP with respect to SHM 1.0. Specifically, the set of weighting functions of QP(22,22) is used by the other QP settings within the delta QP=0 and the set of weighting functions of QP(22,24) is used by the other QP settings within the delta QP=2

	AI HEVC 2x		**AI HEVC 1.5x**	
	MPWIP	Test Case	MPWIP	Test Case
Class A	-0.7%	-0.5%	N/A	
Class B	-1.1%	-0.9%	-0.5%	-0.4%
Min.	-0.3%	-0.2%	-0.1%	0.0%
Max.	-2.4%	-2.2%	-1.5%	-1.3%
Overall (EL+BL)	-1.0%	-0.8%	-0.5%	-0.4%

Table 8: Comparison performance of Y-BD-rate savings of an experiment in Test Group B vs. MPWIP with respect to SHM 1.0. Specifically, all the sets of weighting functions in delta QP = 0 are used by QP setting in delta QP = 2 with a constraint that the QP settings have the same QP value of the BL share the same set of weighting functions

	AI HEVC 2x		**AI HEVC 1.5x**	
	MPWIP	Test Case	MPWIP	Test Case
Class A	-0.7%	-0.5%	N/A	
Class B	-1.1%	-0.8%	-0.5%	-0.4%
Min.	-0.3%	-0.2%	-0.1%	0.0%
Max.	-2.4%	-2.1%	-1.5%	-1.3%
Overall (EL+BL)	-1.0%	-0.7%	-0.5%	-0.4%

Table 9: Performance of the QP setting unification with respect to SHM 1.0

		AI HEVC 2x			**AI HEVC 1.5x**		
		Y	U	V	Y	U	V
Class A	Traffic	-0.7%	-0.4%	-0.3%	N/A		
	PeopleOnStreet	-0.5%	-0.5%	-0.2%			
Class B	Kimono	-0.3%	-0.2%	-0.2%	-0.1%	-0.1%	-0.1%
	ParkScene	-0.2%	-0.1%	-0.2%	0.0%	-0.1%	-0.1%
	Cactus	-0.9%	-1.0%	-1.1%	-0.1%	-0.3%	-0.3%
	BasketballDrive	-2.2%	-2.2%	-1.9%	-1.2%	-1.3%	-1.0%
	BQTerrace	-1.2%	-1.2%	-1.3%	-0.3%	-0.4%	-0.4%
Overall (EL+BL)		-0.8%	-0.8%	-0.8%	-0.3%	-0.4%	-0.4%

Table 10: Performance of the Horizontal and Vertical unification with respect to SHM 1.0

		AI HEVC 2x			**AI HEVC 1.5x**		
		Y	U	V	Y	U	V
Class A	Traffic	-0.7%	-0.4%	-0.3%	N/A		
	PeopleOnStreet	-0.6%	-0.6%	-0.4%			
Class B	Kimono	-0.3%	-0.3%	-0.3%	-0.1%	-0.2%	-0.2%
	ParkScene	-0.2%	-0.2%	-0.2%	0.0%	-0.2%	-0.2%
	Cactus	-0.9%	-1.1%	-1.2%	-0.1%	-0.5%	-0.5%
	BasketballDrive	-2.2%	-2.5%	-2.1%	-1.3%	-1.4%	-1.2%
	BQTerrace	-1.2%	-1.5%	-1.7%	-0.3%	-0.4%	-0.5%
Overall (EL+BL)		-0.9%	-0.9%	-0.9%	-0.4%	-0.5%	-0.5%

Table 11: Performance of MPWIP-U (the combination of two unifications) with respect to SHM 1.0

		AI HEVC 2x			**AI HEVC 1.5x**		
		Y	U	V	Y	U	V
Class A	Traffic	-0.6%	-0.3%	-0.3%	N/A		
	PeopleOnStreet	-0.5%	-0.4%	-0.1%			
Class B	Kimono	-0.1%	-0.2%	-0.2%	-0.1%	0.0%	0.0%
	ParkScene	-0.1%	0.0%	-0.1%	0.0%	-0.1%	0.0%
	Cactus	-0.7%	-0.9%	-1.0%	-0.1%	-0.2%	-0.3%
	BasketballDrive	-2.2%	-2.1%	-1.8%	-1.2%	-1.3%	-1.1%
	BQTerrace	-1.1%	-0.9%	-1.0%	-0.2%	-0.3%	-0.4%
Overall (EL+BL)		-0.8%	-0.7%	-0.6%	-0.4%	-0.4%	-0.4%

from any QP setting within a delta QP group, and then constrained all other QP settings (within the QP group) to use this set and observed the changes of bit rate savings compared to when each QP setting has its own set of weighting functions. In the latter experiment, all the weighting functions obtained within one delta QP group will be used by another group with a constraint that the QP setting sets which have the same QP value at the BL will share the same set of weighting functions. Let’s call all the experiments in the unifying weighting functions within the delta QP setting case as “Test Group A” and all the experiments in the latter case for the unifying weighting functions between the delta QP groups as “Test Group B”. All the possible test cases in both “Test Groups” were examined; however, only one test case in each group is presented in this section, as further described in the caption of Table 7 and Table 8.

As can be seen from Table 7 and Table 8, the Y-BD-rate savings in both cases varies insignificantly between the two unifications and the proposed design. In particular, the bit rate savings drops at most by only 0.3% in all of the cases. Therefore, this provides an opportunity to perform the simplification for weighting functions of the QP settings within a delta QP group and/or for those of the QP settings between two delta QP groups. As a result, the following sub sections are arrived at to support the proposal for the first simplification mode by unifying the weighting functions.

Unifying Weight Tables for Different QP Settings

From the above analysis, we proposed the simplification in which there is only one set of weighting functions for all the QP settings. Table 9 presents the coding performance of the proposed scheme with only one set of weight tables for all QP settings. In this case, the number of weight tables is reduced to only 105. As before, the least-squares method is employed to compute the unified weights for different QP settings. From Table 9 and by comparison with the results in Table 4, a 0.2% decrease in Y-BD-rate savings results, declining from 1.0% and 0.5% to 0.8% and 0.3% in the AI-2x and AI-1.5x test cases, respectively.

Unifying Weight Tables for Horizontal and Vertical Modes

We also observed that the weighting functions for the Horizontal and Vertical modes are related to each other mainly through a transpose operation, which is intuitively agreeable considering the symmetric properties of signal statistics in the natural images. Thus, the other simplification we made is to unify their weight tables. That is, for MPWIP, we apply the same weight tables to the Horizontal and Vertical modes, and the tables only need to be transposed for one of them prior to the application. As a result of this unification, the number of weight tables is reduced to 616. Table 10 shows the resulting coding performance from this simplification. As can be seen and expected, the coding gain in the AI-2x case is reduced slightly from 1.0% to 0.9% (and in the A1-1.5x from 0.5% to 0.4% respectively) and a similar performance decline is observed for both luminance and chrominance components.

The previous two unifications can be applied simultaneously to gain further reduction in memory requirements. The results in Table 11 show that the coding loss due to the application of both simplifications is moderate as compared to the design without any unification of weight tables. In this case, a coding loss of 0.2% and 0.1% was observed in the Y-BD-rate of AI-2x and AI-1.5x, respectively. The total number of weight tables is however reduced from 840 to 77, which amounts to a decrease in storage space.

4.3.2. Constrained MPWIP

With respect to the observation that the weighting functions of the DC and AC components of the same layer have similar waveforms (although their magnitudes can differ considerably), in this section we propose a *Constrained MPWIP*, in which the weight values to associate with the DC and AC components from the same layer are restricted to be identical or, alternatively, the texture information of the EL intra predictor and the reconstructed BL block are to be weighted. There is an additional constraint; that the weight values for these aforementioned texture information must

Table 12: Performance of the Constrained-MPWIP with respect to SHM 1.0

		AI HEVC 2x			AI HEVC 1.5x		
		Y	U	V	Y	U	V
Class A	Traffic	-0.2%	-0.4%	-0.4%	N/A		
	PeopleOnStreet	-0.2%	-0.5%	-0.4%			
Class B	Kimono	-0.2%	-0.2%	-0.2%	0.1%	0.0%	-0.1%
	ParkScene	-0.2%	-0.1%	-0.2%	0.0%	-0.2%	-0.2%
	Cactus	-0.4%	-0.8%	-0.8%	-0.1%	-0.5%	-0.4%
	BasketballDrive	-1.7%	-1.1%	-1.0%	-1.0%	-0.3%	-0.3%
	BQTerrace	-0.4%	-0.6%	-0.8%	-0.1%	-0.4%	-0.4%
Overall (EL+BL)		-0.5%	-0.5%	-0.6%	-0.2%	-0.3%	-0.3%

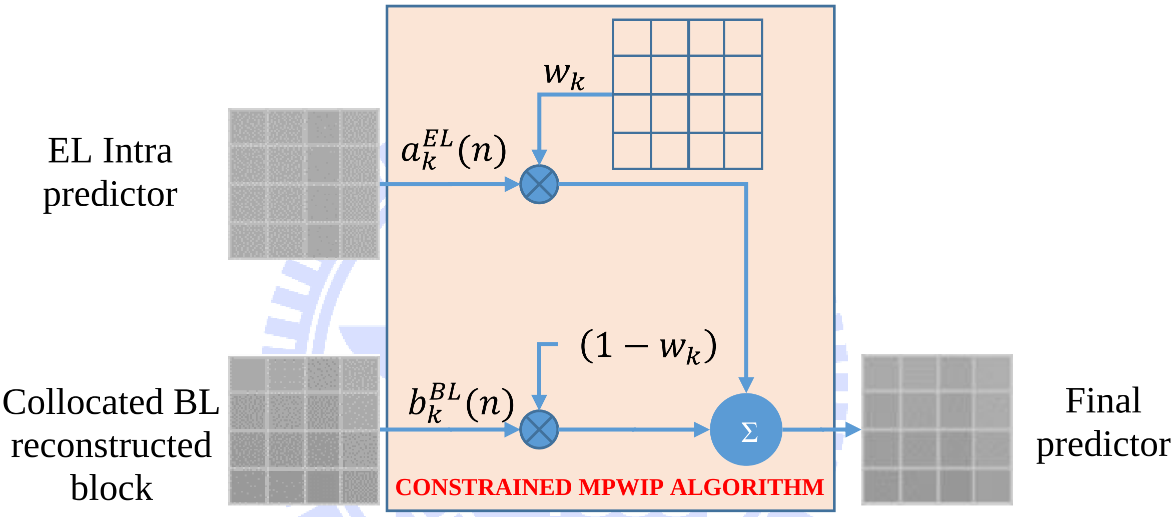
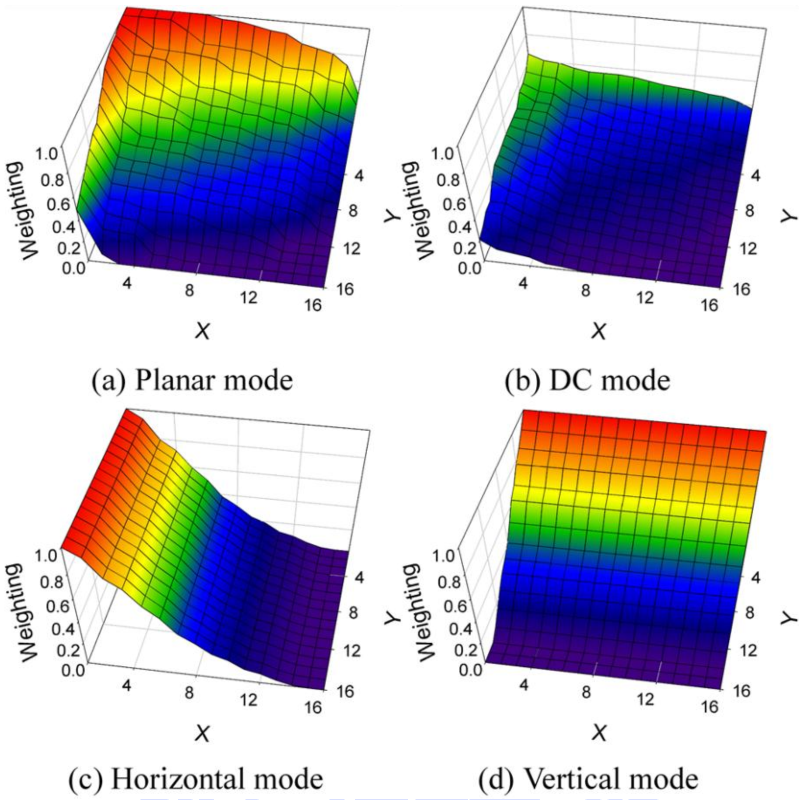


Figure 17: The Constrained MPWIP Scheme

add up to one; the constraint is referred to as the *unit-gain constraint* [9]. The use of this constraint is due to the fact that each of these textures can be considered as a prediction candidate; therefore, for the worst case, the sum of these textures could be blown out to the range of values that predefined the sample. Essentially, this *Constrained MPWIP* becomes one that forms a prediction of the EL by linearly combining the reconstructed BL block and the EL (directional) intra predictor with an adaptive weighting at the pixel level, as illustrated in Figure 17, and that is therefore similar to the scheme proposed in [6]. Furthermore, it is observed that all the unifying



(a) Planar mode

(b) DC mode

(c) Horizontal mode

(d) Vertical mode

Figure 18: Waveform of weighting functions for all test modes at block size of 16x16, QP(30,30). (a) Planar mode, (b) DC mode, (c) Horizontal mode, (d) Vertical mode

techniques found in the previous section can be applied to this scheme. Therefore, when viewed as a simplified version of MPWIP, this scheme can be considered as an extension of the MPWIP-U scheme and requires only 21 weight tables, one (rather than four as in the proposed design) weighting function for each EL prediction mode, only a single set of weight tables for all QP settings, unifying weight tables for the Horizontal and Vertical modes and for different block sizes which are the luma/chroma type. As an example, Figure 18 depicts the weighting function for the EL intra predictor produced with four test modes for the luminance component at the block size of 16x16.

Table 12 shows the coding performance of the constrained MPWIP, as compared to the SHM-1.0 anchor. As expected, it incurs a moderate coding loss of 0.5% and

0.3% in the AI-2x and AI-1.5x cases, respectively, when compared to the original design. This provides beneficial separation of signals into DC and AC components in forming a better EL intra predictor.

In an attempt to examine the effect of the *unit gain constraint* in terms of bit rate savings, the supplementary experiments were carried out with all the test conditions being the same as those of the *Constrained MPWIP*, except that the '*unit gain constraint'* was relaxed. It can be concluded that the bit rate savings of this supplementary experiment varies inconsiderably compared with the 'Constrained MPWIP'. In addition, the supplementary experiment has a higher cost in terms of increasing the number of weight tables to twice as many.

Conceptually, the *Constrained MPWIP* scheme is very similar to the proposed WIP algorithm in which the texture information of the BL and the EL are combined adaptively according to the pixel's position. Therefore, it is beneficial to make a comparison between the *Constrained MPWIP* and the WIP algorithm. From Table 6 and Table 12, it can be seen that our simplified mode provides a 0.5% coding gain for the AI-2x and 0.3% for AI-1.5x cases respectively, while the WIP algorithm achieved 0.3% and 0.1% gains respectively. There are two major differences between our scheme and the WIP algorithm that enables our algorithm to provide a higher gain: 1) we created four additional modes and the best mode for each block is determined by the RDO process, while the WIP algorithm is applied to all intra prediction modes; 2) it seems that the WIP algorithm uses a unified weighting scheme for all the intra prediction modes, even though there are cases where it is not necessary to weight some neighboring reference pixels in the EL; in our mode, we have different weighting functions for each intra mode, which leads to a more appropriate weighting scheme to form the final prediction. However, each mode in our scheme has its own weighting functions and the number of weight tables also depends on the prediction block size, so that the cost of our scheme is higher in terms of memory storage to store those weight tables.

4.3.3. Summary

The findings of this chapter would be summarized as follows

- It turns out that the performance of the proposed algorithm (MPWIP) depends strongly on the characteristics of the video sequences. Specifically, the proposed scheme works best in sequences that contain more homogeneous regions.
- For the *Unification to MPWIP* simplifications, it can be concluded that the number of weight tables is reduced considerably with moderate R-D losses, compared to MPWIP.
- For the *Constrained MPWIP*, even though the number of weight tables can be further reduced, the losses seem to be significant, compared to MPWIP. Therefore, it justifies the benefits of separating the texture information into the AC and DC components.

CHAPTER 5

Conclusions

5.1. Summary

In this thesis, we have introduced a sophisticated algorithm to combine the EL intra predictor and the BL reconstructed block targeted to improve the EL intra prediction in the framework of the TextureRL; and this algorithm does bring a coding gain. The algorithm first separates those textures into the AC and DC components, and then weights them by different weight values; the weight value to associate with each component is a function of the prediction pixel's position in the block. In addition, the parameters (e.g. the intra prediction mode of the EL intra predictor, the QP setting, and the prediction block size) that affect the weighting scheme were thoroughly analyzed. From those analyses, the important observations can be summarized as follows:

1. The weighting functions associated with each AC and DC component of both layers are dependent on the prediction mode of the EL intra predictor,

2. The effect of the QP setting on the weighting functions is insignificant in terms of bit rate savings in the common test conditions, and
3. The weighting functions depend quite significantly on the prediction block size in which the higher weight values are given to the weighting functions associated with the EL components in the smaller block size.

However, the simulation results reveals that the coding gain from the proposed design is rather limited, and may not justify the significant increase in computational complexity and memory requirement to store the weight tables. In addition, though its simplified versions (e.g. Unification to MPWIP simplifications, and Constrained-MPWIP) can reduce the memory requirement, the coding gains are also reduced considerably. Furthermore, a similar observation was also made for other tools adopting the TextureRL approach [11]. Therefore, it may not be worthwhile to construct the scalable extension to HEVC in the TextureRL framework.

5.2. Future Works

Instead of applying the proposed weighting scheme to the AC and DC components, more sophisticated components derived from the texture information can be considered. Those components can be extracted from the texture information by different types of filtering methods. Therefore, a more thorough analysis on the frequency components of the texture information can be taken into account as an extension of this thesis.

The other aspect that we have not fully explored in this thesis is whether the costly floating-point operation that occurs in our weighting scheme is necessary. It can perhaps be replaced by a simpler fixed-point operation; however, the impact on coding performance remains to be investigated.

Bibliography

[1] E. Alshina, A. Alshin, and J. Park, “TEA2: Cross-check of RefIdx performance (tests 3.2.1),” *ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JCTVC-L0090*, Jan. 2013.

[2] G. Bjontegaard, *Calculation of Average PSNR Differences between RD Curves*, ITU-T SG16/Q6 Doc. VCEG-M33, Apr. 2001.

[3] B. Bross, W.-J. Han, J.-R. Ohm, G. J. Sullivan, and T. Wiegand, “WD4: Working draft 4 of high-efficiency video coding,” *ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JCTVC-F803*, July. 2011.

[4] X. Cao, C. Lai, Y. Wang, L. Liu, J. Zheng, and Y. He, “Short distance intra coding scheme for high efficiency video coding,” *IEEE Trans on. Image Processing*, vol. 22, no. 2, pp. 790–801, Feb. 2013.

[5] J. Chen, J. Boyce, Y. Ye, and M. M. Hannuksela, *SHVC Test Model 1 (SHM 1)*, documents JCTVC-L1007, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), Jan. 2013.

[6] C. K. Kim, “Non-CE1: Weighted intra prediction,” *ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JCTVC-M0117*, Apr. 2013.

[7] J. Lainema and K. Ugur, “TE3: Results of test 4.1.2 on intra DC correction,” *ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JCTVC-L0036*, Jan. 2013.

[8] X. Li, J. Boyce, P. Onno, and P. Onno, “Common test conditions and software reference configurations for the scalable test model,” *ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JCTVC-L1009,* Jan. 2013.

[9] M. Orchard and G. Sullivan, “Overlapped block motion compensation: an estimation-theoretic approach,” *IEEE Trans on. Image Processing*, vol. 3, no. 5, pp. 693–699, 1994.

[10] H. Schwarz, D. Marpe, and T. Wiegand, “Overview of the scalable video coding extension of the H.264/AVC standard,” *IEEE Trans on. Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, Sept. 2007.

[11] G. Sullivan and J.-R. Ohm, *Meeting report of the 13th meeting of the Joint Collaborative Team on Video Coding (JCT-VC), Incheon, KR, 18-26 Apr. 2013,* documents JCTVC-M Notes, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), Apr. 2013.

[12] G. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, “Overview of the high efficiency video coding (HEVC) standard,” *IEEE Trans on. Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[13] X. Xiu, Y. He, Y. He, and Y. Ye, “TEA2: Inter-layer reference picture placement,” *ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JCTVC-L0051*, Jan. 2013.

[14] J. Zan, “TE2: Results of test 3.1.3 on CU based Intra-BL signaling,” *ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JCTVC-L0165*, Jan. 2013.