Improved Techniques for Video Compression and Communication

Haoming Chen

A dissertation

submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

University of Washington

2016

Reading Committee:

Ming-Ting Sun, Chair

Eve Riskin

Jenq-Neng Hwang

Program Authorized to Offer Degree:

Electrical Engineering

University of Washington

Abstract

Improved Techniques for Video Compression and Communication

Haoming Chen

Chair of the Supervisory Committee:
Professor Ming-Ting Sun
Department of Electrical Engineering

Video compression and communication has been an important field over the past decades and critical for many applications, e.g., video on demand, video-conferencing, and remote education. In many applications, providing low-delay and error-resilient video transmission and increasing the coding efficiency are two major challenges.

Low-delay and error-resilient video coding is critical for real-time two-way video communication over wireless networks. Intra-refresh coding, which embeds intra coded regions into inter frames can achieve a relatively smooth bit-rate and terminate the error propagation caused by the transmission loss. In this dissertation, we propose a novel linear model for the intra-refresh cycle-size selection adapting to the network packet loss rates and the motions in the video content. We also propose a strategy that adapts the intra-refresh coding pattern to different

cycle-size and obtains better R-D (Rate-Distortion) performance compared to traditional intra-refresh patterns. Experimental results show that the linear cycle-size selection model can achieve about 3dB improvement compared with a fixed cycle-size. Also, with the proposed intra-refresh order, a 2.0% bitrate reduction is obtained in average compared with the vertical-partition intra-refresh.

The Intra Block Copy (IntraBC) is a newly adopted tool in the H.265/HEVC (High Efficiency Video Coding) extension for the screen content video coding. The IntraBC tool efficiently encodes repeating patterns in a picture. The current IntraBC scheme achieves about 1.0% bit-rate reduction on average and up to 4.3% bit-rate reduction on natural content video for a database consisting of 2K, 4K, and 8K sequences. In this dissertation, we propose to improve the IntraBC with a template matching block vector and a fractional search IntraBC. With these two tools, the gain on natural content video coding can be further improved by 0.5% on average and up to 2.0%.

Intra prediction is an important tool in intra-frame video coding to reduce the spatial redundancy. In current coding standard H.265/HEVC, a copying-based method based on the boundary (or interpolated boundary) reference pixels is used to predict each pixel in the coding block to remove the spatial redundancy. We find that the conventional copying-based method can be further improved in two cases: 1) the boundary has an inhomogeneous region, and 2) the predicted pixel is far away from the boundary that the correlation between the predicted pixel and the reference pixels is relatively weak. This dissertation performs a theoretical analysis of the optimal weights based on a first-order Gaussian Markov model and the effects in these two cases. It also proposes a novel intra prediction scheme based on the analysis that smoothing the copying-based prediction can derive a better prediction block. Both the theoretical analysis of the

coding gain and the experimental results show the effectiveness of the proposed intra prediction method. An average gain of 2.3% on all intra coding can be achieved with the H.265/HEVC reference software.

Noise affects the efficiency of motion compensated prediction and degrades the coding performance. Some techniques, e.g., bi-directional prediction, have been utilized to decrease the effects of the noise. In this dissertation, we propose to further reduce noise and improve compression performance by applying a bilateral filter in the motion compensated prediction. The bilateral filter can remove the noise while preserving the edges. Our experimental results show that adding the bilateral filter can achieve a 4.9%, 1.6%, 1.6% average gain, in H.265/HEVC Low-Delay P (LDP), Low-Delay B (LDB), and Random Access (RA) coding, respectively, over a set of HD and UHD sequences up to 8K resolution.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

AI: All Intra

AVC: Advanced Video Coding

BD-Bitrate: Bjøntegaard Delta Bitrate

BD-PSNR: Bjøntegaard Delta PSNR

BV: Block Vector

CABAC: Context Adaptive Binary Arithmetic Coding

CU: Coding Unit

DCT: Discrete Cosine Transform

HEVC: High Efficiency Video Coding

IntraBC: Intra Block Copy

GOP: Group of Pictures

LCU: Largest CU

LDB: Low-Delay B

LDP: Low-Delay P

MB: Macroblock

MC: Motion Compensation

MOS: Mean Opinion Score

MPEG: Moving Picture Experts Group

MV: Motion Vector

MSE: Mean Square Error

PLR: Packet Loss Rate

PSNR: Peak Signal-to-Noise Ratio

QP: Quantization Parameter

RA: Random Access

R-D: Rate-Distortion

RDO: Rate Distortion Optimization

SAD: Sum of Absolute Difference

SSD: Sum of Square Difference

SSIM: Structural Similarity

ACKNOWLEDGEMENTS

First and foremost I offer my sincerest gratitude to my advisor, Prof. Ming-Ting Sun, for the continuous support of my Ph.D. study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this dissertation. I could not have imagined having a better advisor for my Ph.D. study.

Besides my advisor, I would like to thank the rest of supervisory committee: Prof. Eve Riskin, Prof. Jenq-Neng Hwang and Prof. Brian Curless, for their time and effort serving on my committee and reviewing my dissertation.

I take this opportunity to thank all the collaborators in my research for their valuable suggestions as well as my fellow colleagues and mentors during my internships for their various kinds of help. I would like to thank my lab-mates for the discussions on research and all the fun we have had in the past four years.

I would like to thank my parents for their love and support. Last but not the least, I would like to thank my wife Rui, for her understanding, support and encouragement.

DEDICATION

To my family

Chapter 1. INTRODUCTION

Video compression and communication has been an important field over the past decades and critical for many applications, e.g., TV broadcasting, video on demand, videoconferencing, and remote education. Several standards have already been developed for video coding, including MPEG-2 [1], MPEG-4 [2], H.264/AVC [3], H.265/HEVC [4] and Google's VP9 [5]. It is pointed out in [6] that video accounted for about 66% of the consumer Internet traffic in 2013 and will be about 79% in 2018. With ubiquitous mass market applications, many techniques have been proposed to improve the quality of the compressed video, make the compressed video more error-resilient, and reduce the transmission delay.

Due to the importance and practical significance of video coding, a large number of companies and research labs have been working on improving the coding performance. The current state-of-the-art video coding standard is H.265/HEVC. With the optimization effort over the past decades, now it is becoming very challenging to be able to achieve even a small coding gain over H.265/HEVC. For example, in some recent publications in the IEEE Transactions on Circuit and Systems for Video Technology and IEEE Transactions on Image Processing, 1 or 2% of average improvements over H.265/HEVC are considered significant [7]–[10]. Nonetheless, due to the importance, many research organizations are still actively working on improving the coding performance. It is expected that with the accumulation of small improvements it will eventually result in overall significant improvements. A new video coding standard activity on next generation video coding is expected to begin soon.

In this dissertation, we propose new techniques for video compression and communication. These methods are discussed in Chapter 2 to Chapter 5, and are briefly summarized as follows.

Chapter 2 focuses on the low-delay [11], [12] error-resilient video coding, utilizing a method called intra-refresh coding, which embeds intra coded regions into inter frames. This chapter presents a new linear model for the intra-refresh cycle-size selection adapting to the network packet loss rates and the motions in the video content. It also analyzes issues in designing the intra-refresh coding pattern and the refresh order, and proposes a strategy which can adapt to different cycle-size and obtain better Rate-Distortion (R-D) performance compared with traditional random intra-refresh and vertical-partition intra-refresh. Experimental results show that the linear cycle-size selection model works effectively, where a 3dB improvement can be achieved compared with a fixed cycle-size. Also, with the proposed intra-refresh order, a 2.0% bitrate reduction is obtained in average compared with the vertical-partition intra-refresh [13], [14].

Chapter 3, Chapter 4, and Chapter 5 focus on improving the coding efficiency of the video codec beyond H.265/HEVC. Intra Block Copy (IntraBC) is a newly adopted coding tool in the H.265/HEVC extension for the screen content video coding. The IntraBC tool efficiently encodes repeating patterns in a picture. The current IntraBC scheme achieves up to 3.4% and 1.0% on average bit-rate reduction on natural content video for a database consisting of 2K, 4K, and 8K sequences. In Chapter 3, we propose to improve the IntraBC with a template matching block vector and a fractional pixel search IntraBC. With the IntraBC and these two tools, the gain on natural content video coding can be up to 5.4% and 1.5% on average [15].

Chapter 4 discusses the intra prediction in video coding. In current coding standards (e.g. H.264 and H.265/HEVC), a weighted linear combination of the reference pixels is used to predict each pixel in the coding block to improve the prediction accuracy. This chapter performs a theoretical analysis of the optimal weights based on a first order 2D Gaussian Markov model and the effect when the pixel values deviate from the model. This chapter also proposes a novel intra

prediction scheme based on the analysis. Both the theoretical analysis and the experimental results show the effectiveness of the proposed intra prediction method. An average gain of 2.3% can be achieved with the modified H.265/HEVC reference software.

Chapter 5 discusses the inter prediction in video coding. Noise affects the efficiency of motion compensated prediction and degrades the coding performance. Some techniques, e.g., bi-directional prediction, have been utilized to decrease the effects of the noise. In this chapter, we propose to further reduce noise, and improve compression performance by applying a bilateral filter [16] in the motion compensated prediction. The bilateral filter can remove the noise while preserving the edges. Our experimental results show that adding the bilateral filter can achieve a 4.9%, 1.6%, 1.6% average gain, in H.265/HEVC Low-Delay P (LDP), Low-Delay B (LDB), and Random Access (RA) coding, respectively, over a set of HD and UHD sequences up to 8K resolution.

In Chapter 6, we conclude this dissertation and discuss some possible future works.

Chapter 2. ADAPTIVE INTRA-REFRESH FOR LOW-DELAY ERROR-RESILIENT VIDEO CODING

2.1 BACKGROUND

Low-delay and error-resilient video coding is critical for real-time video chat applications over wireless networks [17]–[21]. On the delay aspect, given a fixed network throughput, the delay induced by the video codec mostly depends on the video encoder buffer size. On the error-resilience aspect, the error induced by the packet loss will propagate to the subsequent frames due to the nature of the predictive coding framework. To terminate an error-propagation, conventional video coding schemes encode an intra-coded frame (I-frame) in each Group-Of-Pictures (GOP) [1]. An I-frame is encoded without any reference to other frames, and so, it is not affected by errors in the previous frames. For a GOP with a size of K frames, in the worst case when the beginning I-frame is lost, error propagation is limited to within the following K-1 Predictive frames (P-frames), assuming for low-delay applications the B-frames (Bi-directional predicted frames) are not used. However, for the low-delay aspect, the GOP coding structure may not be the best choice. Since an I-frame only exploits the spatial redundancy within itself, it generates much more bits than a P-frame. The resulting bit-stream usually needs to be smoothed by a relatively large encoder buffer to ensure it does not exceed the network transport capability. This rate-smoothing encoder buffer could cause relatively long delay. Since the I-frame appears periodically, this delay is induced not only at the initial stage, but also in the following video transmission.

Note that besides the insertion of I-frames, some other error-resilient schemes including robust entropy coding [22], [23], forward error correction (FEC) [24], and unequal error protection [25] have been proposed. In [26], some error-resilient tools used in H.263 and MPEG-4 are

reviewed. A more recent review of the error-resilient coding tools is in [27]. This paper focuses on the error-resilient intra-prediction, which is more relevant to our work. Some techniques mentioned in [27] are also discussed in this section.

Intra-refresh (or intra-slice) coding schemes can provide low-delay and good error resilience features. In intra-refresh coding, instead of encoding the whole frame as an I-frame, a subset of Macroblocks (MBs) in each frame can be forced into intra-coded MBs, so that after a cycle of frames, the whole frame is completely refreshed. This spreading of intra MBs into the P-frames can produce a quite uniform bit-rate. With the relatively uniform bit-rate, the encoder buffer could be avoided or kept to a minimal size to result in a low-delay video codec. Besides offering low-delay, intra-slice coding schemes also provide good error-resilience performance. Since each frame is completely refreshed after a cycle of frames, the parts of the picture affected by the transmission errors will be constantly refreshed. A vertical partition intra-refresh scheme as shown in Figure 2.1 is applied in x264 [28], a popular open-source software, encoding videos into the H.264/MPEG-4 AVC format. In the vertical partition intra-refresh scheme, given the intra-refresh cycle-size of *N* frames, the whole frame is split into *N* regions vertically. In the mode decision process, the MBs within each intra-refresh region (shaded area in Figure 2.1) are forced to be intra-coded.

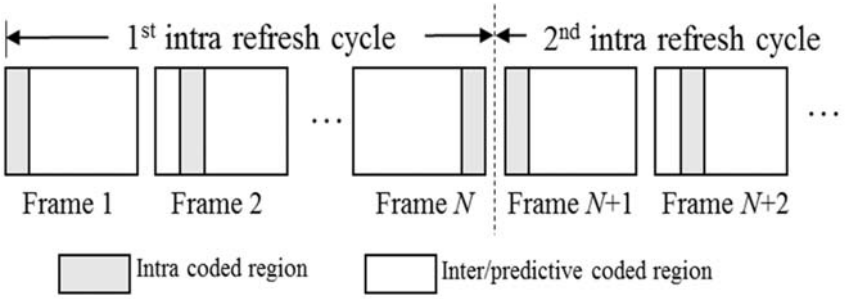


Figure 2.1. Vertical partition of intra-refresh regions.

To achieve a better performance under different network conditions, a fixed cycle-size intra-refresh scheme is not optimal. In [29], a rate-distortion model is proposed considering the channel (network packet loss rates) and the source (the intra-coded MB percentage) jointly. Based on this model, given the network condition, the optimal number of intra-coded MBs can be derived. However, some empirical and sequence-dependent parameters make the derivation of the optimal intra-coded MB percentage (and thus the number of frames in an intra-refresh cycle) difficult.

Besides the selection of the number of frames in an intra-refresh cycle, intra-refresh coding can be applied to MBs in different grouping patterns, resulting in different rate-distortion performance. One method is to choose the intra-refresh MBs randomly, which is the method used in [29], [30], and the default intra-refresh method in the latest JM reference software JM18.6 [31]. In the random intra-refresh, the percentage of MBs refreshed in each frame is set to $1/N$. To avoid duplicated intra-refreshing, refreshed MBs are tracked so that one MB is not refreshed twice within one intra-refresh cycle. In this way, it guarantees that all MBs are refreshed once in a cycle. Another method based on Rate-Distortion (R-D) cost (considering the packet loss rate) of each Macroblock was proposed in [32] to improve the coding efficiency. However, the random or R-D-based MB selection may cause dislocation artifacts [33] when errors occur next to the intra-coded parts. Moreover, the scattered distribution of intra-coded MBs decreases the compression efficiency due to the constrained intra prediction: to make the intra-coded region not be affected by the error propagation from the inter-coded region, the intra-prediction pixels cannot come from the neighboring inter-coded region [34]. An attention-based adaptive intra-refresh coding scheme is proposed in [33] which employs an attention area (or region of interest) extraction algorithm and applies the intra-refresh on these grouped attention MBs. This scheme shows good subjective

quality of the transmitted video over an error-prone network. However, the scheme does not cover the whole picture and errors out of the attention area may propagate for a long time. An isolated-region-based method is proposed in [35]. In the high packet loss-rate scenario, this method works well. However, it has lower coding performance when the packet loss-rate is low, since the prediction efficiency is not considered.

Some cyclic intra-refresh algorithms that can cover the whole frame are proposed in [36]–[40]. To guarantee a complete and efficient error recovery, the refreshed region in a cycle cannot predict from an unrefreshed part, otherwise the error may be propagated into the refreshed part and the intra-refresh scheme becomes ineffective. This protection principle for the refreshed region restricts the potential motion compensation candidates and cause a compression efficiency drop, especially when the refresh direction (e.g., left to right in Figure 2.1) is opposite to the motion direction in the frames. This restriction affects the bitrate significantly. A bad intra-refresh order could increase the total bitrate by as much as 10% compared with a good intra-refresh order [36]. Considering this problem, a motion adaptive intra-refresh scheme is proposed in [36]. In this scheme, every frame is split into 3×4 partitions and the refresh order of these 12 regions is obtained by training based on the motion in each partition region. This scheme can reduce the motion vector restriction effectively, but to be adaptive to the network packet loss rates, more refresh patterns and different cycle-sizes are needed.

In this chapter, we propose a new intra-refresh coding scheme. In the first part, a content-based linear cycle-size selection model is proposed. Given the packet loss rate and the motion information, the best intra-refresh cycle-size is estimated. Compared with [29], our model is much simpler and has no empirical and sequence-dependent parameter. Also, since our model is a simple linear model, it gives better insights about how the video content affects the cycle-size. It also

relates the model parameters with the content and the motion of the video. All parameters in our model are trained from a set of real sequences and verified on a different set of testing sequences. In the second part, the design of intra-refresh region partition and refresh order is analyzed. Based on considerations in intra-refresh coding, we propose a rectangular partition for intra-refresh. Moreover, we modify and simplify the refresh order patterns in [36] so that they can be applied to different cycle-sizes. Experimental results show that our cycle-size model is effective and the proposed intra-refresh order outperforms the vertical cyclic intra-refresh scheme. Compared to the motion-adaptive intra-refresh scheme in [36], the proposed intra-refresh order has the flexibility that it can be applied to different cycle-sizes.

The rest of this chapter is organized as follows. In section 2.2, a linear model is proposed for the adaptive selection of the cycle-size. In section 2.3, the design of the intra-refresh partition and the order are analyzed. Section 2.4 shows the simulation results. Section 2.5 concludes this chapter.

2.2 Adaptive Content-Based Cycle-Size Selection

Figure 2.2 compares different refresh cycle-sizes N (N = 4, 8, 16, and 32) under different packet loss rates ($10^{-4} \sim 10^{-1}$). "Foreman" and "Mother-daughter" are encoded by x264 with the vertical partition intra-refresh scheme [28]. The sequences are encoded with four different cycle-sizes (4, 8, 16 and 32). Each random packet loss causes a frame loss. The loss frame is concealed by repeating the previous frame. A smaller cycle is better for recovering from the errors more quickly when the packet loss rate is high, and a large refresh cycle is better for a low packet loss rate since it gives better coding gains with a smaller intra-coded area in each frame. Thus, for optimal performance, the refresh cycle-size N should be adaptive to the packet loss rates.

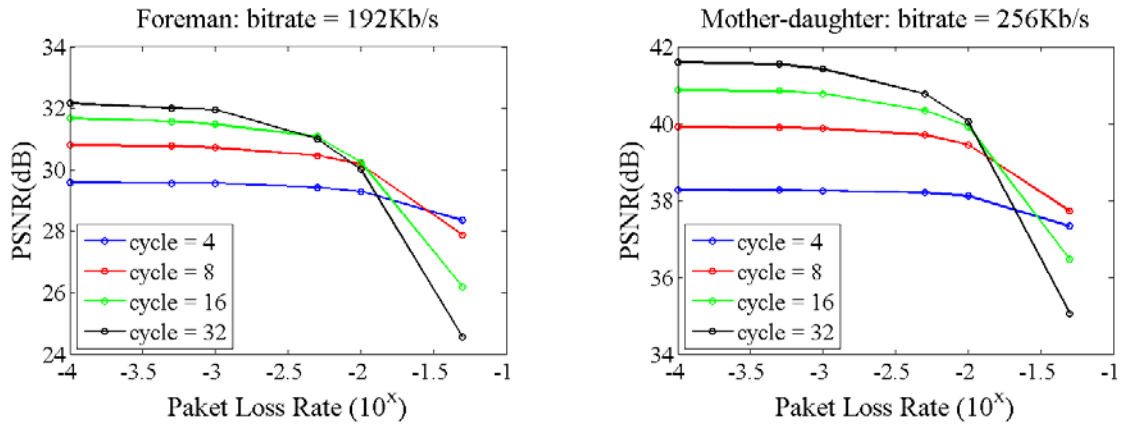


Figure 2.2. Simulation results for the test sequences "Foreman" and "Mother-daughter"

In this section, we proposed a linear model to predict the optimal intra-refresh rate based on the network packet loss rate and the motion in the video content.

2.2.1 *A Linear Model for the Intra-Refresh Rate Based on the Packet Loss Rate*

In [29], a joint end-to-end distortion model is proposed w.r.t. the intra-refresh rate and the packet loss rate:

$$D_s(R_s,\beta) = D_s(R_s,0) + \beta(1-\lambda+\lambda\beta)\times[D_s(R_s,1) - D_s(R_s,0)] \tag{2.1}$$

$$D_c = \frac{a}{(1-b+b\beta)}\frac{p}{1-p}E[F_d(n,n-1)] \tag{2.2}$$

In Eq. (2.1), D_s is the source distortion, which is a function of the source bitrate R_s and the intra-refresh rate β (= number of intra-coded MBs/total number of MBs) in one frame. λ is a sequence-dependent parameter. The source distortion is a linear combination of two extreme cases: all MBs inter-coded $D_s(R_s,0)$ and all MBs intra-coded $D_s(R_s,1)$, both are measured in MSE (Mean Square Error). In (2.2), D_c is the channel distortion caused by the packet loss rate p. Parameter a and b are model constants. $E[F_d(n,n\text{-}1)]$ is the expectation of the difference of the neighboring frames n and n-1, which is also measured by the MSE.

The total end-to-end distortion is the sum of the source distortion and the channel distortion:

$$D = D_s + D_c. \tag{2.3}$$

Under a bitrate R_s and the packet loss rate p, the best intra-refresh rate that minimizes the end-to-end distortion can be derived by taking the derivative of D w.r.t. β and setting it to zero, resulting in a cubic equation of β,

$$A\beta^3 + B\beta^2 + C\beta + D = H\frac{p}{1-p}, \tag{2.4}$$

where

$$A = 2\lambda b^2, \tag{2.5}$$

$$B = b(b + 4\lambda - 5\lambda b), \tag{2.6}$$

$$C = 2(1-b)(b + \lambda - 2\lambda b), \tag{2.7}$$

$$D = (1-b)^2(1-\lambda), \text{ and} \tag{2.8}$$

$$H = \frac{abE[F_d(n,n-1)]}{D_s(R_s,1) - D_s(R_s,0)}. \tag{2.9}$$

We observe that the intra-refresh rate β is the inverse of cycle-size N, so β is always much smaller than 1, e.g., $N = 12$ means $\beta = 0.083$. Also, the coefficients A, B, C, and D in Eqs. (2.5) - (2.9) are constants. Hence, we propose to omit the higher order items and simplify Eq. (2.4) into a linear model as follow:

$$\beta = H'\frac{p}{1-p} + D', \tag{2.10}$$

where $H' = H/C$ and $D' = -D/C$, which are constants (dependent on the sequences). From Eq. (2.10) the optimal refresh rate is linear to $p/(1-p)$.

To verify that the proposed linear model is sufficiently accurate, we test 21 video sequences as shown in Figure 2.3. We encode each sequence at the 1 Mbps bitrate with the cycle-size N from 4 to 40. Note that since the model is independent to the order of the intra-refresh pattern (random, vertical, or motion adaptive in [36]), we use the random intra-refresh pattern. We simulate the packet loss rate p from 0.1% to 20%. Each packet loss scenario is simulated 50 times, the PSNRs of a decoded sequence in each simulation is denoted as PSNR(p, N, i), where i = 1, …, 50. For each p, we select the N that maximizes the average PSNR, where the average PSNR[1] and the best N are defined as follows:

$$\overline{PSNR}(p,N) = \frac{1}{50}\sum_{i=1}^{50} PSNR(p,N,i), \text{ and} \tag{2.11}$$

$$N^*(p) = \arg\max_N \overline{PSNR}(p,N). \tag{2.12}$$

We plot the best β (i.e., $1/N^*(p)$) versus $p/(1-p)$ in Figure 2.3. From the testing results and the fitting lines, we can see that the linear model works well.

In Figure 2.3, the x-axis is $p/(1-p)$, where p is from 0.1% to 20% and the y-axis is the best β, i.e., 1/(Best Cycle-Size), which is in the range of [1/40, 1/4]. Here the "Best Cycle-Size" is the cycle-size (between 4 and 40) that minimizes the distortion for each packet loss rate. Note that for some sequences (e.g., bus, city, crew …), the best cycle-size reaches the smallest cycle-size (limited to 4, β=0.25) when the $p/(1-p)$ is less than 0.25, and then the line becomes flat. We have removed that flat part to show the linearity clearly. The black dashed line is the fitting linear function. We can see that the linear model can approximate the relationship between β and $p/(1-p)$ well.

[1] Here the PSNR of each frame is calculated first and then averaged over the whole sequence. We note that another metric is to measure the mean square error (MSE) of the whole sequence first and then calculate the PSNR of the whole sequence.

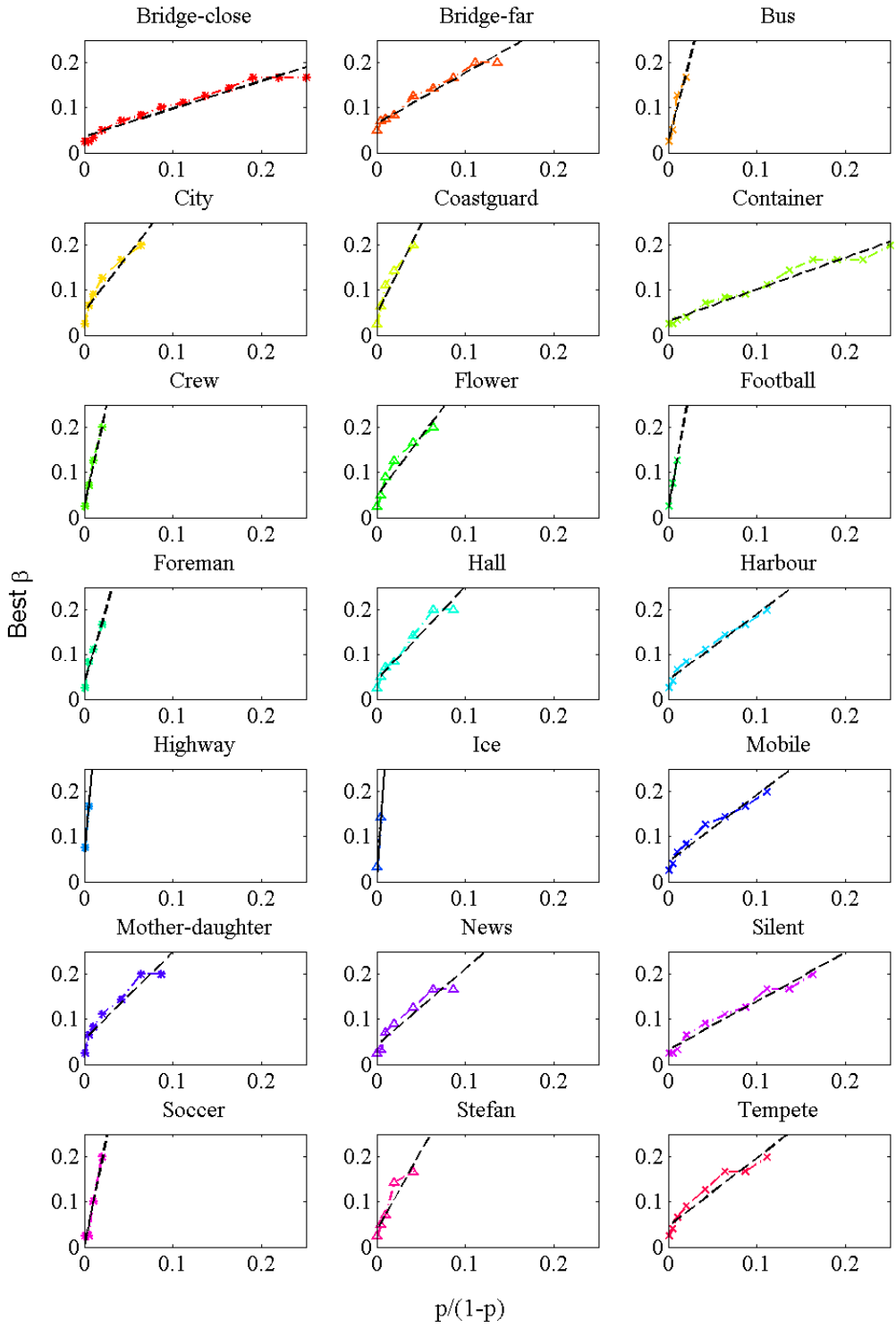


Figure 2.3. Linear model fitting results on 21 sequences.

2.2.2 *Parameters of the Linear Model*

With this linear model, given the parameters *H'* and *D'* we can estimate the best cycle-size easily. However, since *C*, *D,* and *H* depend on many empirical constants, it is difficult to get a close-form solution of *H'* and *D'*. To solve this problem, we plot the 21 corresponding fitting lines in Figure 2.4. We have following observations on these lines: There is a rough relationship between the slope of the linear model and the motion of video content. For example, the Ice and Highway sequences have very fast motion and they have the largest slopes; Container and Bridge-close sequences have very slow motion and have the smallest slopes.

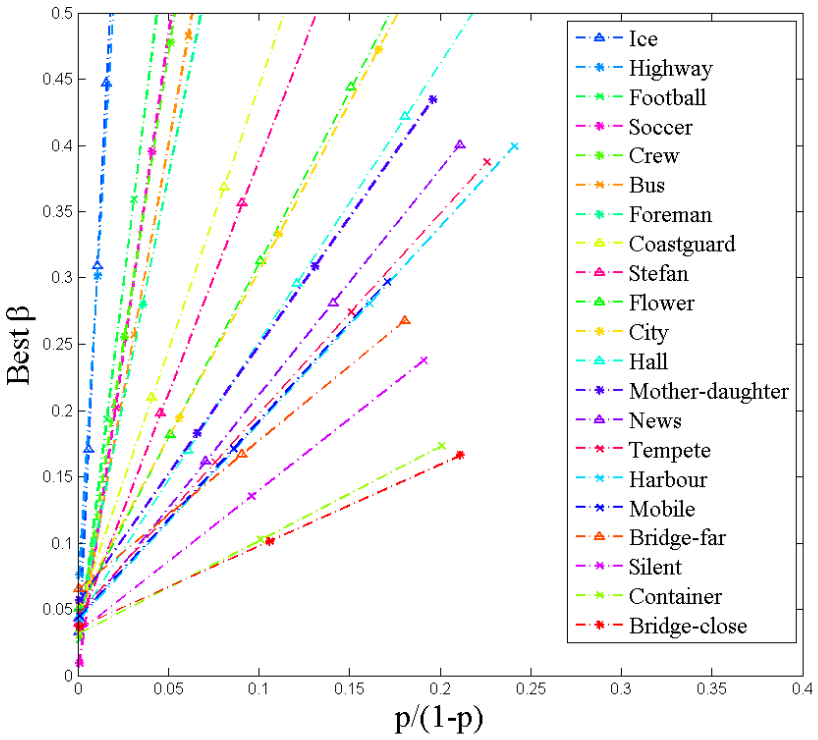


Figure 2.4. Fitting lines of the 21 sequences in the same plot. (Best viewed in color.)

Actually, these observations can be justified from the expressions of C, D and H in Eqs. (2.7) - (2.9). For example, H', which equals to H/C is directly proportional to $E[F_d(n,n\text{-}1)]$ and inversely proportional to $D_s(R_s,1) - D_s(R_s,0)$. $E[F_d(n,n\text{-}1)]$ is the difference of neighboring frames, which is large when the motion of the video content is fast. Moreover, for a fast motion video, the error propagates quickly, so a large intra-refresh rate (or a small cycle-size) is needed. For $D_s(R_s,1) - D_s(R_s,0)$, the distortion difference between all-intra and all-inter under the same bitrate, if it is large, it means force-intra coded MBs will induce more distortion, so a small intra-refresh rate is preferred, resulting in a smaller slope in the linear model.

Based on the above analysis, we model the slope proportional to $E[F_d(n,n\text{-}1)]/[\ D_s(R_s,1) - D_s(R_s,0)]$. To see the proportion, in Figure 2.5, we plot the slope value H' versus $E[F_d(n,n\text{-}1)]/[\ D_s(R_s,1) - D_s(R_s,0)]$.

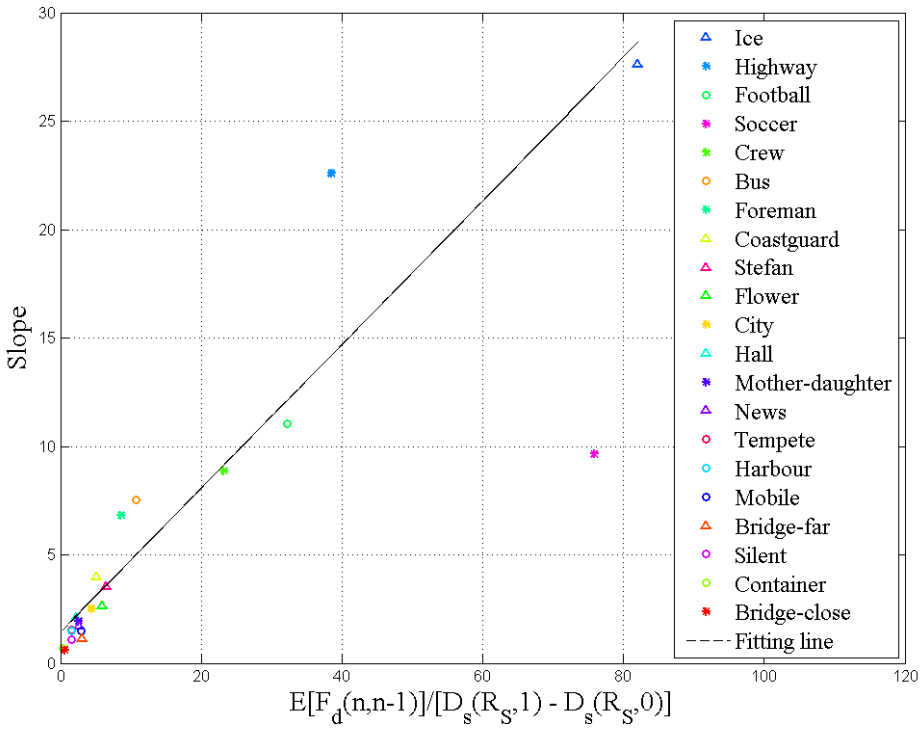


Figure 2.5. The slopes of the linear models versus the proportion of $E[F_d(n,n-1)]$ and $D_s(R_s,1) - D_s(R_s,0)$. (Best viewed in color.)

For each sequence, $E[F_d(n,n\text{-}1)]$ is the average MSE of all neighboring frame pairs and the $D_s(R_s,1) - D_s(R_s,0)$ is the average MSE difference between two cases: all intra coding and all inter coding structures. Then, we fit the points with a linear model, which results in the following model:

$$H' = c \times E[F_d(n,n\text{-}1)]/[\, D_s(R_s,1) - D_s(R_s,0)] + d. \tag{2.13}$$

To verify the model, we separate these 21 sequences into two sets: 11 sequences for training the parameters in Eq. (2.13) and the other 10 sequences for testing the performance of our proposed cycle-size selection model. From the training part, the parameters $c = 0.3164$ and $d = 1.6625$.

For the offset points (D' in Eq. (2.10)) of each line in Figure 2.4, they have very close positions. So in the linear model, we just use the average value of the offsets in the training set as the offset of the proposed model.

2.2.3 *Content-based adaptive cycle-size selection*

Combining Eq. (2.10) and Eq. (2.13), a selection model of cycle-size N is as follow:

$$\beta = \frac{1}{N} = \left[\frac{0.3164 \times E[F_d(n,n-1)]}{D_s(R_s,1) - D_s(R_s,0)} + 1.6625 \right] \times \frac{p}{1-p} + 0.0342. \tag{2.14}$$

However, in low-delay video coding, we cannot get the $E[F_d(n,n\text{-}1)]$, the all-intra MSE $D_s(R_s,1)$, and all-inter MSE $D_s(R_s,0)$ of the whole sequence. One method is to estimate these values based on the first N cycles. However, this may not work when the video content changes. Another method, which is more accurate, is to re-calculate these values every several frames and update them adaptively. In this work, for the convenience of comparing with other schemes (random intra-refresh and vertical partition intra-refresh), we use the former method.

2.3 DESIGN OF THE MB INTRA-REFRESH REGION PARTITION AND ORDER

As mentioned, the distribution and the refreshing order of intra-coded MBs could affect the R-D performance significantly. In this section, we discuss the design of intra-refreshing region partition and the refresh order. We consider the dislocation effect, constrained intra prediction, and constrained inter prediction in our designing process.

2.3.1 *Consideration of Issues in the Intra-Refresh Pattern Design*

Some issues related to the design of intra-refresh patterns are briefly summarized as follows.

A. The dislocation problem

Visual artifacts are caused by the scattered distribution of intra-coded MBs when packet loss occurs. To overcome this problem, in each frame, the forced intra-coded MBs should be grouped together instead of being separated. Some visual results of the dislocation problems are shown in Section 2.4.

B. Constrained intra prediction

The constrained intra-prediction requires that the reference samples of the intra prediction cannot come from the neighboring inter-coded blocks. This constraint can protect the intra-refreshed region so that it is not affected by the potentially corrupted inter-coded regions. In [27], it shows that with the constrained intra prediction, the decoded video can achieve up to about 4 dB gain when the packet loss-rate is between 2% ~ 8%. Note that the constrained intra prediction has been included in H.264/AVC standard. Scattered distribution of the intra-coded regions will decrease the compression efficiency due to the constrained intra prediction, which is an error-resilience scheme. The loss of the coding efficiency due to this constrained intra prediction is shown in Figure 2.6. We tested 5 sequences (all are 352×288, 4:2:0, CIF), For each R-D curve pair

of 5 sequences the dashed line is the R-D curve coded without the constrained intra prediction, the solid line is the R-D curve coded with constrained intra prediction. The BD-bitrate calculation [41] shows that contrained intra prediction will increase the bitrate by 5.1% in average, which is significant.

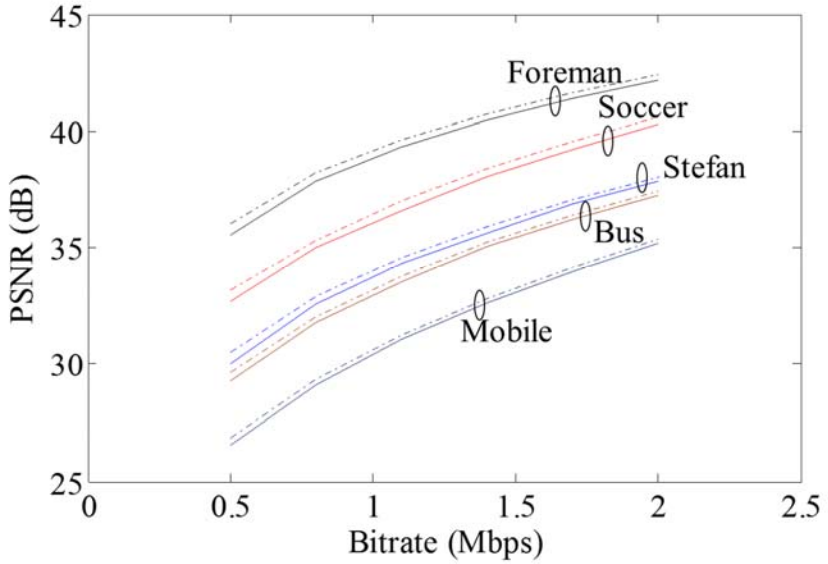


Figure 2.6. Comparison of R-D performance of random intra-refresh coding schemes with and without constrained intra prediction in JM 18.6 [31].

C. Constrained inter prediction

The constrained inter-prediction requires that the reference samples of the inter prediction cannot come from the unrefreshed regions. As a result, the sub-optimal motion prediction could cause a coding efficiency drop. As mentioned in [36], taking the motion of the frame into consideration can reduce such kind of restriction. Note that video standards only specify those parts necessary for interworking, not those parts that only affect the quality. Since intra-frame coding cannot refer to other frames, it is part of the H.264/AVC. The constrained inter prediction

is to prevent the error from propagating from the un-refreshed part. It only affects the quality, thus, it is not part of the standard.

From the above considerations, we can see the design of intra-refresh coding is twofold: one is to design the partition method, which can reduce the dislocation problem and the constrained intra prediction problem; the other is to design the refresh order of these partitioned regions, so that the constrained motion vectors can be reduced.

2.3.2 *Design of Intra-Refresh Region Partition for Reducing the Intra Prediction Constraint*

To overcome the dislocation problem and to make full use of the intra-coded blocks in the intra prediction, grouping the intra-coded blocks together is desired. Given a number M of MBs to be intra-coded in the current frame, they can be grouped in different shapes, e.g., vertical strip, horizontal strip, rectangle (if $M = X \times Y$, where X and Y are integers larger than 1), or other irregular shapes. In intra prediction, a coding block gets its prediction samples from the left and top MBs. However, due to the constrained intra prediction, coding blocks along the top and left boundaries of these grouping shapes can only have partial intra-prediction pixels, so the prediction is less efficient. It should be noted that the longer top and left boundaries, the more blocks have partial prediction pixels. It can be shown that grouping the intra-refreshed MBs into a rectangular shape as close to a square as possible could minimize the total length of the constrained boundaries, and make better use of the intra-coded pixels.

Compared with irregular shapes, grouping intra-refreshed MBs into a rectangle also has an advantage that it is easy to partition one frame into different regions without any overlaps and holes. A straightforward way is to partition a frame into $w \times h$ rectangles, where $w \times h = N$, and w and h are the numbers of partitions along the width and height of the video frame, respectively. When selecting w and h, two factors are taken into consideration:

1) For each N, there may exist more than one partitions. Take N=18 as an example, it can be split into 18×1, 1×18, 6×3, 3×6, 2×9, and 9×2. Based on the grouping principle that the grouped rectangle should be as close to a square as possible, we select the partition leading to the smallest difference between N_w/w and N_h/h, where N_w and N_h are the numbers of MBs along the width and height of the video frame, respectively.
2) For some prime numbers N, e.g., 11, it can only be split into $1\times N$ and $N\times 1$ regions, which conflicts to the grouping principle that the strip shape is less efficient than the rectangle shape close to a square. In this case, we choose the nearest composite number to approximate it, which could be 10 or 12 in this case. The selection of 10 or 12 is based on the same criteria in factor 1).

2.3.3 *Design of Intra-Refreshing Order Reducing the Inter-Prediction Constraint*

With the whole frame split into $w\times h$ rectangles, we design the refresh order so that the constrained inter-prediction effect can be reduced. Since the constrained inter prediction happens on the prediction from the refreshed region in the current frame to the unrefreshed region in the previous frame, we only consider the refresh regions in the current frame. For one refreshed partition region, $P_{n,i,j}$ in the n^{th} frame of one cycle, where $n = 1,\ldots, N$, $i = 1, \ldots, w$, and $j = 1, \ldots, h$, the cost of the constrained inter prediction is approximated by [36]

$$L_{n,i,j} = |\text{mv}_x| + |\text{mv}_y|, \tag{2.15}$$

where mv_x is the horizontal component of the averaging motion vector on region $P_{n,i,j}$ if the horizontal component points to an unrefreshed region, and similarly, mv_y is the vertical component of the averaging motion vector on region $P_{n,i,j}$ if this component points to an unrefreshed region. The total cost is the summation of the cost over all regions and all frames within one cycle,

$$L = \sum_{n} \sum_{i} \sum_{j} L_{n,i,j} \tag{2.16}$$

Since the real motion vectors are not available at the time of selecting the refreshing order of each cycle, the motion vectors of the frame previous to the current cycle is used as an approximation. However, for a cycle-size of *N*, there are *N*! different kinds of refreshing orders and the searching space is so huge that we cannot afford to search every possible refreshing order to find the global optimal order leading to the smallest cost as defined in Eq. (2.16). In [36], 28 different patterns are proposed for the 4×3 partition and it was demonstrated that those patterns can cover most of the camera motions. In our case, since the cycle-size varies, we cannot design a pattern pool specifically for each cycle-size. Based on the simulation results, we observe that using only 6 common patterns, as shown in Figure 2.7, can achieve results almost as good as the 28 different patterns used in [36]. The intra-refresh regions of frames in an intra-refresh cycle are refreshed according to one of the six patterns. We also tried other approaches such as intra-refresh a region connected to the previously refreshed region based on the smallest cost as defined in Eq. (2.16). However, since it could be stuck at local minimum, it is not as good as using one of the six patterns for the whole cycle. Note that these 6 patterns can be extended to different partitions easily. The comparison results are presented in Section 2.4. It should be noted that since the packet loss position could be random, and the regions are refreshed one by one in an intra-refresh cycle, a packet loss could occur at any position in a cycle. Thus, the optimal order of refreshing a particular MB does not depend on where the packet loss occurs, but only depends on the motion due to the constrained inter prediction.

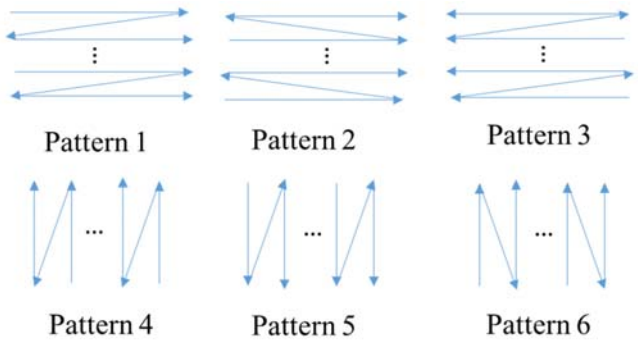


Figure 2.7. Six pattern candidates we evaluated. The intra-refresh regions of frames in an intra-refresh cycle are refreshed according to a pattern shown above.

2.4 EXPERIMENTAL RESULTS

In this section, we first test the proposed design of intra-refresh region partition and the order, and then combine it with the adaptive cycle selection model in Eq. (2.14). In the experiment, we compare our method with randomized intra-refresh (as default in H.264 reference software), the vertical intra-refresh (as default in x264), and the motion-adaptive intra-refresh [36]. The vertical intra-refresh is effective and is the default method in x264. The motion-adaptive intra-refresh can achieve better coding gain than the previous two. We also compare with the optimal cycle size that can achieve the best R-D performance under a certain packet loss-rate.

2.4.1 *Experiments on the Proposed Design of Region Partition and Order*

The experiment is implemented in the latest H.264/AVC reference software JM 18.6 [31]. The test conditions are as shown in Table 2.1.

Table 2.1. Test Conditions

Profile	Main
RD optimization	Yes

Rate control	No
Packet loss	No
Quantization parameters	22, 27, 32 and 37
Number of reference frames	1

Table 2.2 compares the BD-Bitrate savings (a negative number indicates bits saving) on different sequences in source coding (without packet loss) with N=12 and 36. Since in this case, there is no packet loss, our proposed scheme will select a large cycle size N (in this case 36). The comparison with N=12 is just to show that compared with the motion adaptive intra-refresh in [36], although our scheme uses only six refresh-order patterns, it achieves almost the same R-D performance as that in [36] which uses 28 different patterns at N=12. In this table, we can see that our proposed design also outperforms the traditional vertical intra-refreshing under different cycle-sizes. The results under different packet loss rates are discussed in the next part.

Table 2.2. BD-Bitrate Saving (in %) Comparison ("-" Means Gain)

	Our proposed method with $N = 12$		Our proposed method with $N = 36$	
Sequence	v.s. Vertical partition [28] (with $N = 12$)	v.s. Motion adaptive method in [36] (with $N = 12$)	v.s. Vertical partition [28] (with $N = 36$)	v.s. Motion adaptive method in [36] (with $N = 12$)
Bridge-close	-1.5	0.1	-1.1	-17.4
Bridge-far	-5.4	-0.3	-3.6	-24.7
Bus	-0.6	-0.1	0.1	-9.4
City	-4.7	-0.2	-2.3	-18.6
Coastguard	-1.4	0.0	-1.9	-8.8
Container	-3.2	0.4	-1.6	-24.7
Crew	-0.8	0.2	0.2	-4.7
Flower	-4.8	0.0	-2.9	-11.3
Football	-0.4	0.0	-0.2	-3.5
Foreman	-1.5	-0.1	-1.2	-14.0
Hall	-2.0	-0.3	-0.4	-21.3
Harbour	-1.3	0.3	-0.2	-8.9
Highway	-2.5	0.0	-0.8	-9.8
Ice	-0.6	-0.3	-0.1	-7.0
Mobile	-1.0	0.2	-0.5	-11.7

Mother-daughter	-1.6	-0.5	0.1	-22.8
News	-1.9	-0.3	-1.5	-24.5
Silent	-0.7	0.3	-0.7	-25.3
Soccer	-3.4	-0.4	0.2	-4.1
Stefan	-2.1	1.0	0.5	-11.1
Tempete	-1.0	-0.4	-0.2	-9.8
Average	-2.0	0.0	-0.9	-14.0

2.4.2 *Experiments on Adaptive Cycle-Size Selection*

In this test, we use Eq. (2.14) to determine the cycle-size and apply it on the testing set of 10 sequences. All sequences are encoded in 1 Mbps with rate control. The test conditions are listed in Table 2.3, which are different from the previous test.

Table 2.3. Test Conditions

Profile	Main
RD optimization	Yes
Rate control	Yes (1Mbps)
Packet loss	Yes (packet loss rate = 0 ~ 20%)
Error Concealment	Copying the previous image (built-in method in JM18.6)
Number of reference frames	1

The average decoded video PSNRs defined as in Eq. (2.11) are shown in Figure 2.8, where we compare four different cases:

1) Optimal cycle-size that maximizes the PSNR of reconstruction video, using our proposed intra-refresh partition and order.
2) The cycle-size is determined by our proposed selection model Eq. (2.14), using our proposed intra-refresh partition and order.

3) Optimal cycle-size that maximizes the PSNR of reconstruction video, using random intra-refresh coding as default as in JM 18.6.

4) Fixed cycle-size of 12 coding with the scheme in [36].

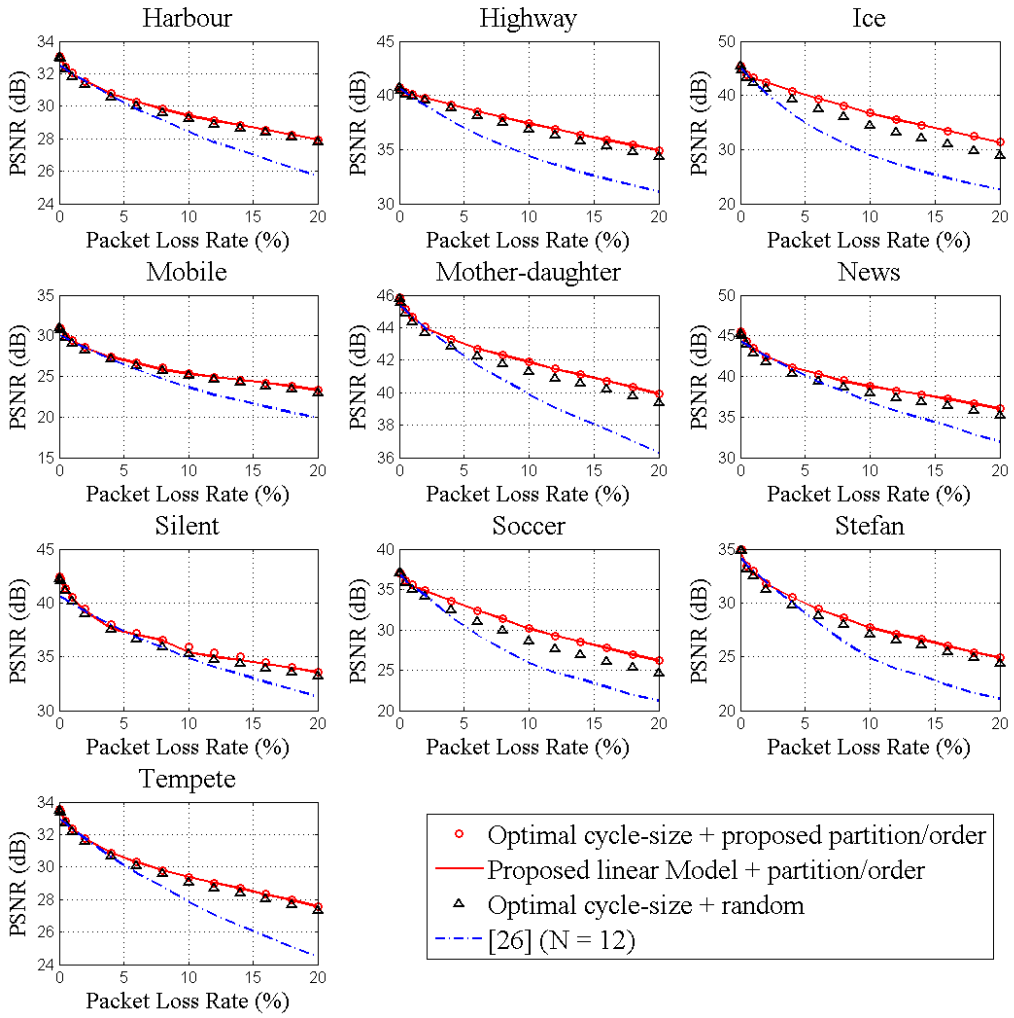


Figure 2.8. Comparison of the proposed cycle-size selection algorithm with our intra-refresh partition and order, optimal cycle-size with our intra-refresh partition and order, the scheme in [36] and the optimal cycle-size with random intra-refresh.

We have following remarks for these results:

1) Our proposed linear model for cycle selection can approximate the optimal cycle-size very well.
2) Combining the cycle selection model and the proposed intra-refresh partition and order, the performance is even better than the optimal results by using the random intra-refresh scheme, especially when the packet loss rate is high (where a small cycle size is used). It is because when the cycle-size is small, the random intra-refresh coding has lower coding efficiency.
3) Compared with [36] with a fixed cycle-size $N = 12$, about 3 dB PSNR gain in average can be achieved especially under a high packet loss rate.

To show the improvement of our proposed method over method in [36] more clearly, we pick up some typical packet loss rates and show the PSNR difference in Table 2.4. Positive number indicates PSNR gain (quality improvement).

Table 2.4. PSNR Gain (in db) Comparison for Some Packet Loss Rates Versus [36]

Sequences	$p = 0.1\%$	$p = 1\%$	$p = 10\%$	$p = 20\%$
Harbour	0.45	0.01	0.98	2.25
Highway	0.00	0.26	2.96	3.82
Ice	0.04	0.65	7.77	8.66
Mobile	0.62	0.11	1.71	3.42
Mother-daughter	0.30	-0.01	1.99	3.59
News	0.69	0.09	1.96	4.06
Silent	1.37	0.52	0.49	2.27
Soccer	0.12	-0.03	4.27	4.97
Stefan	0.52	-0.20	2.86	3.80
Tempete	0.46	0.00	1.53	3.10
Average	0.46	0.14	2.65	3.99

Beside the PSNR, we also compare the decoded video quality between our proposed method and [36] with the SSIM metric [42]. The results are shown in Table 2.5. It is clear that the proposed method also have better quality, especially for high packet loss-rate.

Table 2.5. SSIM Comparison for Some Packet Loss Rate versus [16]

Sequences	p = 0.1%		p = 1.0%	
	[36]	Proposed	[36]	Proposed
Harbour	0.953	0.958	0.947	0.947
Highway	0.954	0.954	0.950	0.951
Ice	0.984	0.984	0.975	0.980
Mobile	0.939	0.945	0.927	0.928
Mother-daughter	0.986	0.986	0.984	0.983
News	0.985	0.987	0.984	0.984
Silent	0.975	0.981	0.973	0.976
Soccer	0.918	0.921	0.900	0.892
Stefan	0.972	0.974	0.957	0.953
Tempete	0.955	0.959	0.950	0.950
Average	**0.962**	**0.965**	**0.954**	**0.954**
Sequences	p = 10.0%		p = 20.0%	
	[36]	Proposed	[36]	Proposed
Harbour	0.895	0.909	0.838	0.880
Highway	0.922	0.936	0.895	0.920
Ice	0.896	0.946	0.826	0.910
Mobile	0.802	0.845	0.664	0.781
Mother-daughter	0.962	0.973	0.938	0.961
News	0.967	0.971	0.947	0.962
Silent	0.948	0.941	0.923	0.927
Soccer	0.744	0.807	0.637	0.733
Stefan	0.802	0.870	0.696	0.801
Tempete	0.886	0.909	0.806	0.874
Average	**0.882**	**0.911**	**0.817**	**0.875**

In practice, the receiver may need to count the packet losses and feedback to the encoder. This process will induce latency between the estimated packet loss-rate and the actual packet loss-rate. To investigate the performance of our proposed method in this scenario, we perform simulations to investigate the PSNR changes with inaccurate packet loss-rate estimations due to the latency of

the feedback from the decoder. In this simulation, a real packet loss rate (1.0%) is used, while the estimated packet loss rate used to derive the intra-refresh cycle size varies within a range (0.1% ~ 4.0%). The results are in Fig. 9. We have following two remarks:

1) For most sequences, the best PSNR appears at the matching point (estimated packet loss rate = actual packet loss rate = 1%). It can further confirm the effectiveness of our proposed linear model.
2) When the estimated packet loss rate deviates from the actual packet loss rate. The PSNR improvement may decrease, but the quality is still relatively good.

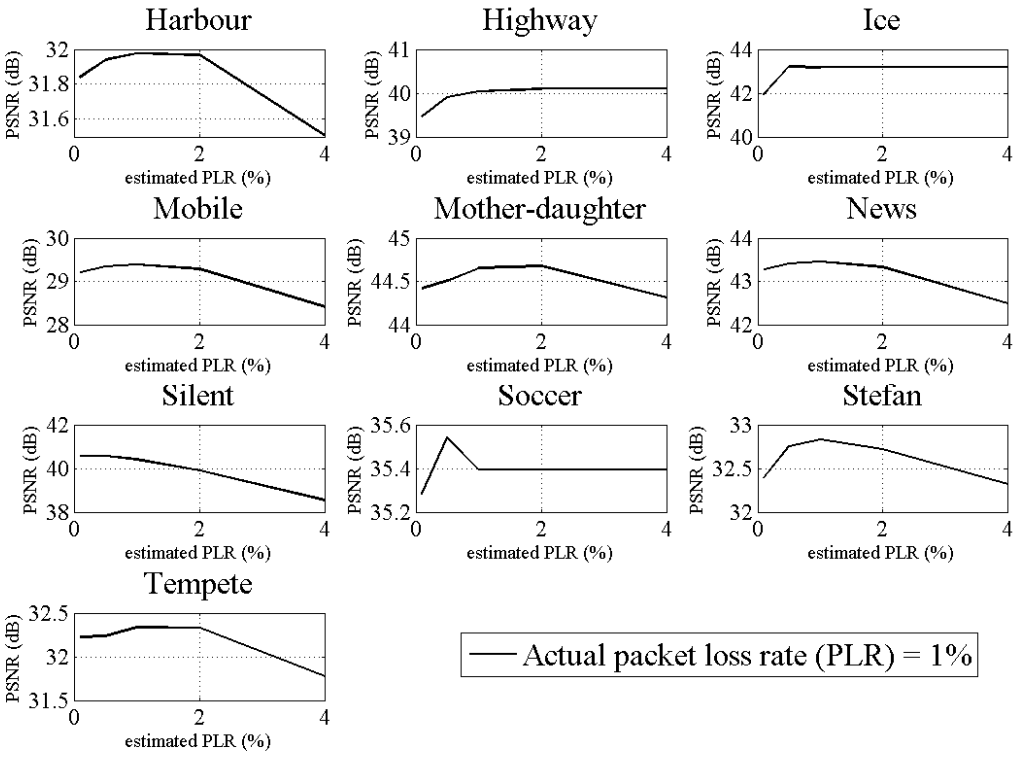


Figure 2.9. Decoded PSNR for different estimated packet loss-rates from 0.1% to 4.0% when the actual packet loss rate is 1%.

2.4.3 *Visual Results*

In Figure 2.10 and Figure 2.11, some visual results of the same frame are compared with different coding methods to show the resultant dislocation artifacts. It is clear that our proposed method have a good error-resilience performance. Compared with the scattered random intra-refresh coding, the dislocation problem is avoided. It should be noted that the MB region groupings in the vertical partition intra-refresh scheme can be considered as a special case of our rectangle region grouping, and so will also not show the dislocation errors. However, due to its partition, some vertical stripe artifacts could be observed, and it has lower coding efficiency (shown in Table 2.4).

(a) Original

(b) w/o intra-refresh (only 1st frame is intra-coded)

(c) w/ random intra-refresh

(d) w/ our partition and refresh order

Figure 2.10. One reconstructed frame from “Foreman” with transmission error. In different methods, the lost frames are the same.

(a) Original

(b) w/o intra-refresh (only 1st frame is intra-coded)

(c) w/ random intra-refresh

(d) w/ our partition and refresh order

Figure 2.11. One reconstructed frame from "Soccer" with transmission error. In different methods, the lost frames are the same.

2.5 CONCLUSION

In this chapter, we present an efficient intra-refresh cycle-size selection model depending on the network packet loss rate and the motion in the video content. We also present a design of the intra-refresh partition and ordering which is based on the dislocation error, constrained intra-prediction, constrained inter-prediction, and the cycle-size. To select the best cycle-size adaptively according to the packet loss rate, we propose a linear model, where no heuristic sequence-specified parameters are needed. Experimental results confirm the effectiveness of the proposed algorithms.

Chapter 3. IMPROVEMENTS ON INTRA BLOCK COPY IN NATURAL CONTENT VIDEO CODING

3.1 BACKGROUND

The Intra Block Copy (IntraBC) mode is a newly adopted extension tool in H.265/HEVC for screen content coding (SCC) [43], [44]. In the IntraBC scheme, a previous encoded block within the frame can be used as a predictor for the current block. This scheme works efficiently on screen content coding since usually there are many repeated patterns in the screen content video, and IntraBC can remove the redundancy much better than the traditional intra prediction (where the predictors only come from neighboring pixels). However, it has been shown that the gain of the current IntraBC scheme on natural content video coding (about 0.5% according to [45] and about 1.0% on a different test set of HD and UHD sequences that we use in our simulation) is much smaller than that on the screen content video. In this chapter, we describe our proposed improvements on the IntraBC scheme to increase the coding performance on natural content videos.

In inter-frame motion estimation, the motion vector is encoded explicitly in the bit stream. To remove the redundancy between neighboring motion vectors, a motion vector predictor is used to generate the motion vector differences. After prediction, the motion vector differences approximately follow the geometry distribution and the Exponential-Golomb coding [46] is used to encode the vector differences. A similar process is applied in the current IntraBC block vector (BV) coding. However, since the neighboring BVs in intra-frame coding may not have high correlations, the vector difference coding still needs a large amount of bits. To overcome this problem, [47] and [48] investigate the use of boundary template matching in H.264/AVC, so that

the BV can be derived from previous reconstructed regions on the decoder side without encoding in the bit stream. It was pointed out that using some boundary matching schemes, the coding gain could be up to 20% in H.264/AVC intra-frame coding. In this chapter, we investigate the use of boundary template matching in the IntraBC for H.265/HEVC.

In the current IntraBC, the prediction block is searched on an integer-pixel level. This is suitable for computer-generated graphics and can reduce the searching complexity and motion vector overhead bits. However, in camera-captured videos, it was pointed out that fractional search can achieve a good gain in inter-prediction [49]. Half-pel search was adopted in H.263 and MPEG-2, quarter-pel search was adopted in MPEG-4 and H.264/AVC, and a 1/12-pel search scheme was proposed in the early stage of H.265/HEVC standard [50]. In this chapter, we investigate the use of fractional search in IntraBC for intra-prediction to increase its prediction accuracy. We also address related issues such as the padding in the interpolation and the entropy coding of the block vectors for the fractional search in the IntraBC.

The reminder of the chapter is as follows: Section 3.2 discusses the template matching BV scheme. Section 3.3 discusses the fractional IntraBC search. Section 3.4 presents experimental results of these two methods. Section 3.5 concludes this chapter.

3.2 Template Matching Intra Block Copy

As described in the introduction, the BVs in the neighboring blocks may not have high correlations, thus the vector differences after prediction are still large, resulting in large overhead bits for encoding the BV differences. Our experimental results in Table 3.1 show that omitting the BV difference bits, the overall BD-bitrate [41] reduction in intra frame coding can be up to 9.0%, which is very significant for the H.265/HEVC standard.

Table 3.1. BD-Bitrate Reduction When Omitting Overhead Bits for The BV Differences. ("-" Means Gain)

Sequences	BD-Bitrate difference
BasketballDrill	-9.0%
BasketballPass	-3.1%
BlowingBubbles	-3.2%
BQMall	-4.1%
BQSquare	-8.8%
PartyScene	-7.1%
RaceHorses	-2.9%
RaceHorsesC	-4.9%

To reduce the overhead bits for the BV differences, there are three possibilities: 1) finding better BV predictors to reduce the BV differences, 2) finding more suitable entropy coding schemes for IntraBC, and 3) deriving the BV on the decoder side so that the transmitted overhead can be completely avoided. Obviously, these three methods can be used jointly. In this section, we focus on 3), and use the boundary matching scheme to implicitly transmit the BV information in the IntraBC.

3.2.1 *Boundary template matching*

We use an inverse-L area adjacent to the coding block as the matching template, as shown in Figure 3.1. In this way, all the matching pixels in both the encoder and the decoder sides are reconstructed prior to the current coding block. For the selection of the template width, we use a 2-pixel width, based on our testing results.

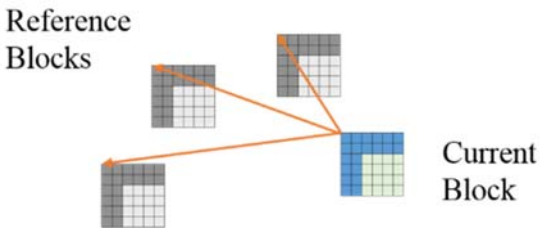


Figure 3.1. Current coding block (green) with its adjacent matching template (blue) and some reference blocks (gray). (Best view in color.)

With the aforementioned template, we can estimate the similarity between the current coding block and the reference candidates based on the matching template difference (SAD, Sum of Absolute Difference, is used in our experiments). It is likely that the reference block is similar to the current coding block when the boundary parts of these two blocks match well.

Obviously, this template matching method is not based on the cost of the actual coding block, and so, the template-matching-based BV may not be as good as the BV found in the traditional block-matching method. To guarantee the performance is not bad in the worst case, we use this template matching BV as an extra mode, instead of replacing all motion-estimated BVs in the IntraBC. Hence, one overhead flag is transmitted to the decoder to indicate if the template matching BV is used.

3.2.2 *Matching template refinement*

To improve the matching, we perform a refinement on a set of N template matching BV candidates with smallest SADs within the search range. Among these N BVs, the index i ($1 \leq i \leq N$) of the vectors with the minimum SADs for the coding block will be transmitted to the decoder side. In the decoder side, the identical set of BV candidates can be derived. After decoding the index i, the i-th candidate will be used as the final BV.

In [47], it is pointed out that using the average of some similar blocks can generate a better predictor. Hence, in this refinement scheme, we use the index *N* to indicate the averaging of all *N* candidates. In our experiments, we select *N* as 8.

3.2.3 *Overhead encoding for boundary matching*

In the proposed template matching scheme, two extra overhead flags are introduced, one is indicating whether the boundary matching method is used, and the other is the refinement index *i*. In our implementation, we use entropy coding to encode the indicating flag, and use 3-bit fixed-length coding for the refinement index. Our experimental results show that using the entropy coding for the indicating flag can reduce the overall encoding bitrate by 0.1%.

3.3 FRACTIONAL INTRA BLOCK COPY SEARCH

In the inter-frame fractional-pel search in the H.265/HEVC reference software HM-14.0 [51], the integer motion vector is derived based on the minimum SAD, and then the half-pel refinement is applied around the integer motion vector position, followed by the quarter-pel refinement based on the half-pel refined position, as shown in Figure 3.2. In the current H.265/HEVC standard, a DCT-based 7 or 8 tap filter is applied for the half and quarter pixel interpolation. The similar fractional search scheme of inter-frame motion estimation can be applied on the IntraBC. Some issues will be discussed as follows.

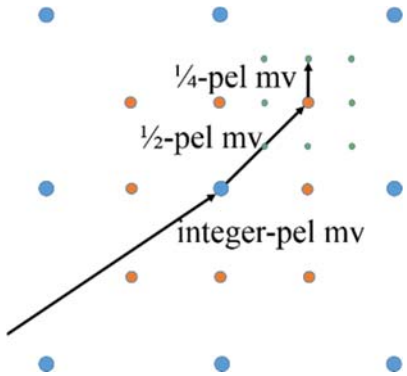


Figure 3.2. Three-step refinement for fractional motion vectors.

3.3.1 *Padding in the interpolation*

Since only the pixels within the current frame and reconstructed prior to the current coding block can be used as the prediction pixels, some pixels used in the 7 or 8 tap interpolation filter may not be available, as shown in Figure 3.3. We investigated two methods to handle this situation: 1) skipping the fractional refinement when those corresponding interpolation pixels are not available, and 2) performing some padding schemes for the interpolation so that each integer position can perform a fractional refinement. Based on simulation results, skipping the refinement on some blocks reduces the overall gain of fractional search; therefore, we choose to implement the padding for the unavailable pixels.

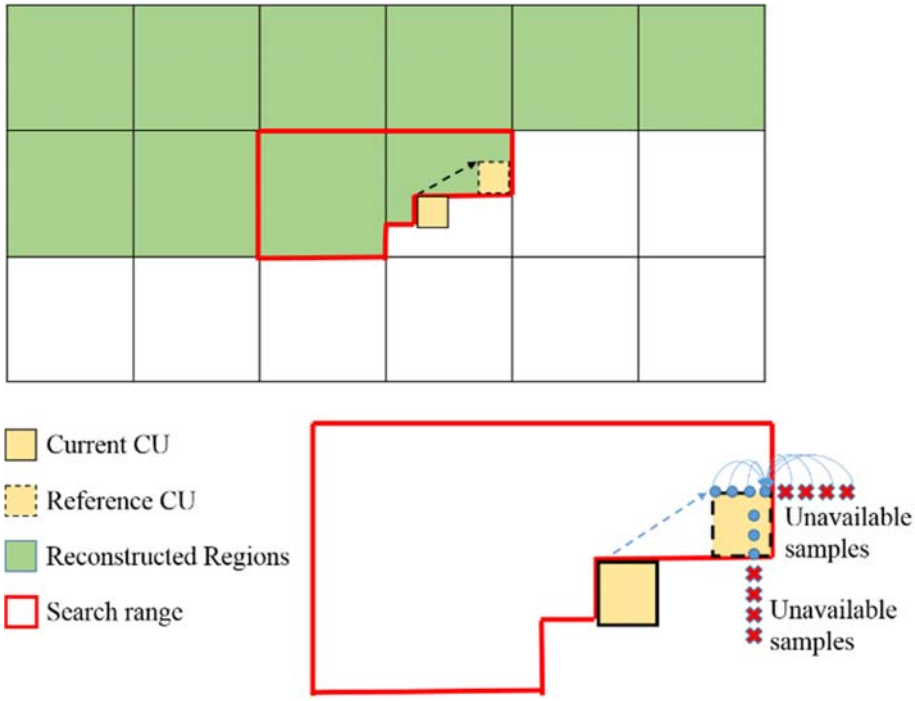


Figure 3.3. IntraBC search range and unavailable samples in half and quarter pixel refinement. (Best viewed in color)

Different padding schemes can be used such as repeating padding or symmetric padding. In the repeating padding, the unavailable pixel is padded with the same value of the nearest pixel which is available on the same row or column. In the symmetric padding, the unavailable pixel is padded with the pixel symmetric to padding boundary. Based on our simulation results, different padding schemes do not affect the coding gain much. Considering the simpler implementation of the repeating padding scheme, we use it in our experimental part. Note that this padding scheme is applied on both luma and chroma components.

3.3.2 *Exp-Golomb coding of quarter-pel BV difference*

Exp-Colomb coding with the order k (≥ 0) is a kind of universal code. The larger k, the closer the code-word is to the fixed-length coding. In the current IntraBC scheme, k is set to 1. However, with a quarter-pel accuracy BV, the BV and the BV difference will be 4 times larger than that with integer-pel accuracy, which increases the bits for BV by 8 bits (4 bits for each X and Y components). There are two methods to encode the fractional BV part (0, 1/4, 1/2 and 3/4) more efficiently. One is to increase k to 3 for the fractional part, which makes the Exp-Golomb coding suitable for fractional BVs. Another way is to encode the fractional BV part separately by a fixed-length coding, since those four parts are almost uniformly distributed. In our simulation, the later method is slightly better, and compared with the current Exp-Golomb coding in the IntraBC, it can reduce the average increment of bits for BV differences from 8 bits to 4 bits.

3.4 EXPERIMENTAL RESULTS

The proposed fractional search and template matching BV schemes are implemented in the H.265/HEVC reference software HM-15.0+RExt-8.0+SCM-2.0 w/o IntraBC (anchor). Some test conditions are shown in Table 3.2. Note that in our simulations, to be consistent for all sequences, we have used 5 frames only, as the computation time for 8K sequences is significant. From our simulations on several frames on lower resolution sequences, the behavior for intra tools can be very well approximated by the results from very few intra frames (one to five).

Table 3.2. Test Conditions

Coding configuration	All-intra coding
QP	22, 27, 32 and 37 (main-tier)
IntraBC search range	Left and current LCU (Largest CU)
Test sequences	4:2:0 YUV sequences with different resolutions (2K, 4K and 8K)

Number of frames	5

3.4.1 *Results of the proposed tools*

To compare the performance of our proposed tools, we perform the following tests:

1) Compare the anchor vs. IntraBC,
2) Compare the anchor vs. IntraBC + "template matching",
3) Compare the anchor vs. IntraBC + "fractional IntraBC", and
4) Compare the anchor vs. IntraBC + "template matching" + "fraction IntraBC".

The BD-bitrate reduction percentages of the above tests are shown in Table 3.3, where a negative number means coding gain.

Table 3.3. BD-bitrate Difference Compared to the Anchor on Different Tests ("-" Means Gain)

		Test 1	Test 2	Test 3	Test 4
2K (1920×1080)	Kimono	-0.0%	-0.1%	-0.1%	-0.1%
	ParkScene	-0.3%	-0.5%	-0.3%	-0.5%
	Cactus	-1.5%	-2.3%	-2.0%	-2.8%
	BasketballDrive	-0.8%	-1.5%	-0.7%	-1.9%
	BQTerrace	-2.2%	-3.7%	-2.2%	-3.8%
4K (3840×2160, except Hotel 4096×2048)	chuno_s4	-0.1%	-0.3%	-0.1%	-0.3%
	chuno_s31	-0.1%	-0.2%	-0.0%	-0.0%
	crownrun	-0.1%	-0.1%	-0.1%	-0.2%
	hotel	-0.9%	-1.7%	-1.4%	-2.3%
	Pku_girl	-0.2%	-0.5%	-0.1%	-0.5%
	reed	0.0%	0.0%	0.0%	0.0%
	4k_seq1	0.0%	-0.1%	0.0%	-0.1%
	butterfly	-0.3%	-0.7%	-0.2%	-0.7%
8K (7680×4320)	8K_seq7	-0.4%	-0.8%	-0.2%	-0.8%
	8K_seq12	-1.1%	-1.6%	-1.2%	-1.8%
	8K_seq14	-4.3%	-4.7%	-3.7%	-4.2%
	8K_seq17	-0.1%	-0.2%	-0.1%	-0.2%
	8K_seq18	-2.2%	-2.8%	-2.0%	-2.8%
	DS_store	-3.4%	-3.8%	-5.0%	-5.4%

	2K average	-1.0%	-1.7%	-1.1%	-1.9%
	4K average	-0.2%	-0.4%	-0.2%	-0.5%
	8K average	-1.9%	-2.3%	-2.0%	-2.5%
	All average	-1.0%	-1.4%	-1.1%	-1.5%

We have following remarks on these results:

1) For tests (2) and (3), the BD-bitrate reduction of template matching is up to 1.5% (BQTerrace) and the BD-bitrate reduction of fractional IntraBC is up to 1.6% (DS_store).
2) The BD-bitrate reduction of combining proposed two tools is up to 2.0% (DS_store). The overall BD-bitrate reduction of the improved IntraBC over H.265/HEVC on natural content video coding is up to 5.4% and 1.5% on average.
3) Comparing tests (2), (3) and (4), it can be seen that for most sequences, the improved gains of fractional IntraBC and template matching are additive.

Considering there is only a small portion of blocks using the IntraBC mode in natural content video coding (most blocks are coded as intra prediction modes), our proposed tools have a significant gain on the IntraBC-coded blocks.

3.4.2 *Hit ratio of our proposed tools*

To demonstrate that our proposed tools on IntraBC is efficient, we plot the hit ratio for the sequence “Hotel” in Figure 3.4. In this figure, the “regular IntraBC” indicates the blocks using integer-pel IntraBC with explicitly transmitted BV. It can be seen that most IntraBC-coded blocks use our proposed tools. More than half of the IntraBC-coded blocks use the template matching BV scheme. This is consistent to the result that the template matching BV scheme has better performance than the fractional IntraBC.

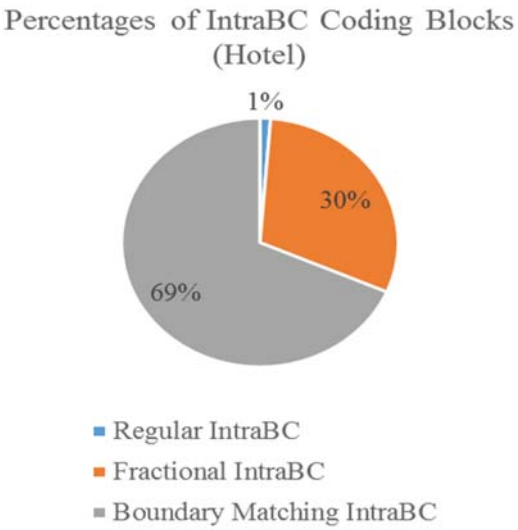


Figure 3.4. Percentage of IntraBC coding blocks using regular IntraBC (the current IntraBC scheme in the H.265/HEVC standard), fractional BV, and boundary matching IntraBC.

The usage of different IntraBC-coded blocks is shown in the frame in Figure 3.5 as an example. It should be noted that the IntraBC blocks often appear at the object boundary. For the flat parts, the IntraBC scheme is not as good as the traditional intra prediction.

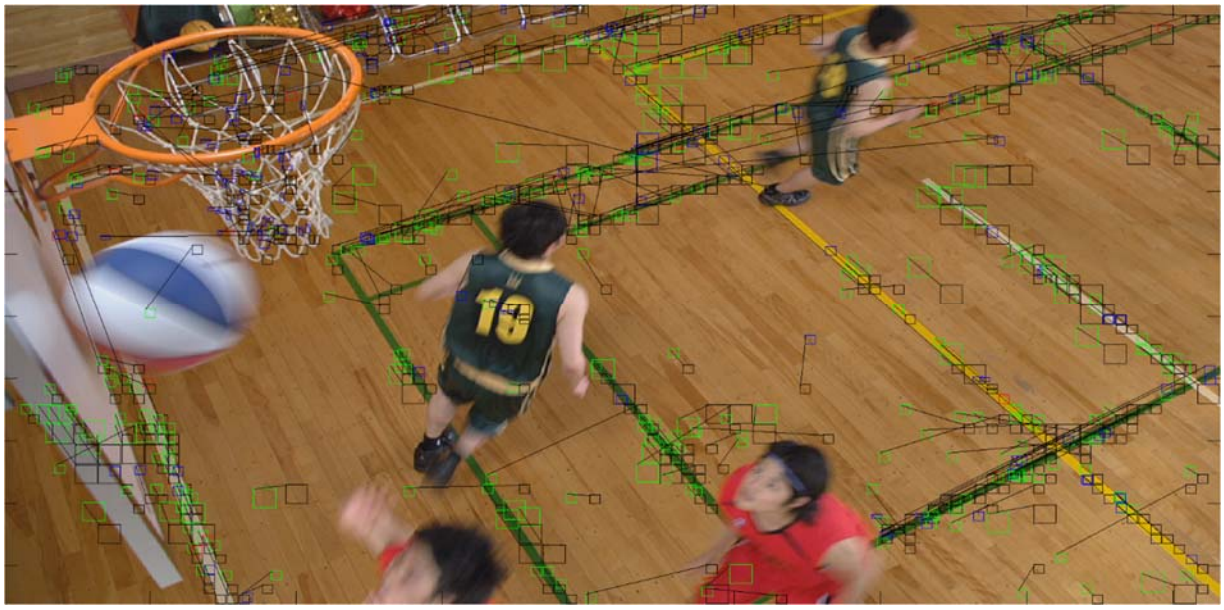


Figure 3.5. Different types of IntraBC-coded blocks in a frame in “BasketballDrill”. Black is the current block being coded. (Blue: Fractional IntraBC, green: Boundary Matching IntraBC, red: Regular IntraBC. Best view in color.)

3.5 CONCLUSION

In this chapter, we present a fractional IntraBC and template matching block vector scheme which is more suitable for natural content coding. With IntraBC and our improvement tools, the gain is up to 5.4% (1.5% on average).

Chapter 4. IMPROVING INTRA PREDICTION IN HIGH EFFICIENCY VIDEO CODING

4.1 BACKGROUND

Intra coding is essential in both still image and video coding. In the block-based coding scheme, the spatial redundancy can be removed by utilizing the correlation between the current pixel and its neighboring reconstructed pixels. From the differential pulse code modulation (DPCM) in the early video coding standard H.261 to the angular intra prediction in the latest H.265/HEVC [52], different intra prediction schemes have been employed. Almost without exception, linear filters are used in these prediction schemes. In H.264/AVC and H.265/HEVC, a copying-based intra prediction is used. In this scheme, a reference pixel (or a filtered one over its neighboring pixels) is copied as the predictor of current pixel.[2] Recently, recursive prediction methods with 3-tap filters [53] or 4-tap filters [54] are proposed to improve the intra prediction accuracy. In both methods, the prediction weights are derived under the first-order 2-D Gaussian Markov model, in which the parameters are trained from real video sequences. In [55], it compares the copying-based intra prediction used in H.264/AVC and the recursive intra prediction, and presents a new training method for the parameters in the first-order 2-D Gaussian Markov model. Fundamentally, the design of linear intra prediction can be formulated as finding suitable prediction weights. It has been shown that given the statistics of the coding blocks as well as the reference pixels, the Wiener filter [56] is the optimum filter leading to the minimum prediction

[2] To avoid confusion, it is noted that the "copying-based intra prediction" is unrelated with the newly adopted "intra block copy" mode in the screen content video coding extension (HEVC-SCC).

mean square error (MSE) [57]. Based on the statistical model trained from real images, a position-dependent filtering intra prediction is proposed in [58]. Each position under different block size and intra prediction mode has its own (Wiener) filtering weights. However, a large table of filtering weights is needed, and it is unsuitable for H.265/HEVC with block sizes up to 64×64 and the number of modes up to 35. A partial-differential-equation-based (or briefly PDE-based) image inpainting technique is applied in the intra prediction [59]. It assumes the image patch is smooth so that the prediction block is generated by solving a PDE problem with known boundary pixels. An improved method using mode-dependent angular masks is proposed in [60]. However, the computation complexity for solving the PDE problem is much higher than the linear intra prediction scheme. In previous linear intra prediction methods, the boundary reference pixels are assumed to have high quality and follow the 2-D Gaussian Markov model. However, in the real sequences, some boundary pixels may not be available, or have noise and/or quantization errors [61].

In this chapter, we first theoretically analyze the optimal prediction assuming the pixels in the video frames follow an underline first-order Gaussian Markov model. It shows that the copying-based method is a good approximation of the optimal prediction when the inter-pixel correlation is high. However, when the pixel is far away from the reference and the correlation is low, it may need a smaller prediction weight. We define the pixels with error as those pixels with pixel values deviate from the underline model. This error will account for the quantization errors, noises, and edge pixels. It shows that the availability and the errors of the boundary pixels affect the intra prediction accuracy. The theoretical coding gain of different intra prediction methods are compared. Based on the theoretical analysis, the chapter presents a novel intra prediction scheme via properly smoothing the copying-based intra prediction.

It should be noted that besides using one-pixel width reference pixels adjacent to the current block, some other intra prediction schemes, e.g., intra block copy [43], [15], [62] and palette coding [63] have been proposed in the screen content coding [64]. Although these methods show a very significant gain in the screen content coding, the gain on the natural content video coding is not as good as that of the conventional linear prediction method. Hence, in this chapter, we focus on the improvements on the linear intra prediction scheme using boundary reference pixels.

The remainder of this chapter is as follows: Section 4.2 discusses the optimal intra prediction when predicted pixels are far away from reference pixels and the reference pixels have error. The theoretical coding gain is analyzed as well. Section 4.3 proposes the novel intra prediction scheme with block-size-dependent iterative filtering and discusses the design of this filter. The experimental results are presented in Section 4.4. Section 4.5 concludes this chapter.

4.2 THEORETICAL CODING GAIN WITH INTRA PREDICTION

The theoretical coding gain comparison is used in the design of transform coding [65]–[68]. This section utilizes the coding gain to discuss different prediction schemes and the effect when the predicted samples are far away from the reference samples and reference samples have error.

4.2.1 *Intra Prediction in 1-D Source without Error*

Consider a 1-D sequence, $X = [x_0, x_1, \ldots, x_N]^T$, following the zero-mean unit-variance first-order Gaussian Markov chain model, its auto-correlation matrix is:

$$E[XX^T] = \begin{pmatrix} 1 & \rho & \rho^2 & \cdots & \rho^N \\ \rho & 1 & \rho & \cdots & \rho^{N-1} \\ \rho^2 & \rho & 1 & \cdots & \rho^{N-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho^N & \rho^{N-1} & \rho^{N-2} & \cdots & 1 \end{pmatrix}, \tag{4.1}$$

where the ρ is the correlation coefficient, which is usually close to 1.

Suppose the first sample x_0 is the boundary value which is used to predict x_i, $i = 1, .. N$, there exists a prediction matrix P generating the predictors $\hat{X}$ as follows:

$$P = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ p_1 & 0 & 0 & \cdots & 0 \\ p_2 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_N & 0 & 0 & \cdots & 0 \end{pmatrix}, \text{ and} \tag{4.2}$$

$$\hat{X} = PX\,, \tag{4.3}$$

where p_i, $i = 1, \ldots, N$, are the prediction weights on x_0 for each sample. Note that only the first column of P has non-zero values, which means the prediction only depends on x_0. The problem of finding the optimum prediction P_{opt} leading to the least predictor error can be formulated as

$$P_{\text{opt}} = \arg\min_{P} E[\sum_{i=1}^{N} (x_i - p_i x_0)^2] \tag{4.4}$$

It is easy to get the optimum p_i by taking the derivatives of Eq. (4.4) with respect to p_i and set it to zero. The solution is

$$p_{i,\,\text{opt}} = E(x_i x_0) \,/\, E(x_0 x_0). \tag{4.5}$$

Substitute the auto-correlation with the values in Eq. (4.1), the optimum prediction weights under the first-order Gaussian Markov chain model is

$$p_{i,\,\text{opt}} = \rho^i,\ i = 1, \ldots, N. \tag{4.6}$$

With optimum prediction matrix, the residuals $Y = [y_0, y_1, y_2, \ldots, y_N]^T$, is derived by $X - PX$

$$Y = (I - P)X, \tag{4.7}$$

where I is the identity matrix. The element in the auto-correlation matrix of Y is

$$E[y_i y_j] = \rho^{|i-j|} - \rho^{i+j}\,,\ i, j = 1, \ldots, N\,. \tag{4.8}$$

The intra prediction process in Eq. (4.7) can be treated as a transform and the coding gain of

a transform is the ratio of the geometric mean of the transform domain sample variance before and after the transform [69]. In the intra prediction case, since the reference samples (e.g., x_0 in the 1-D case) are unchanged, the variance of these samples is excluded in the coding gain calculation. The coding gain defined in decibel is

$$G = 10\log_{10}\left(\left(\prod_{i=1}^{N}\sigma_{x_i}^2\right)^{1/N} \Big/ \left(\prod_{i=1}^{N}\sigma_{y_i}^2\right)^{1/N}\right), \tag{4.9}$$

where the variances of x_i and y_i are the auto-correlation $E(x_i x_i)$ and $E(y_i y_i)$ since they are zero-mean. Substitute the variances with values in Eq. (4.1) and Eq. (4.8), the coding gain with optimal prediction weights for a 1-D first-order Gaussian Markov source is

$$G_{\mathrm{1D,opt}} = -\frac{10}{N}\sum_{i=1}^{N}\log_{10}(1-\rho^{2i}). \tag{4.10}$$

The coding gain of the copying-based intra prediction can be derived with Eq. (4.9) similarly. Note that when N is small, copying x_0, i.e., $p_i = 1$ in Eq. (4.2), is a good approximation of the optimum intra prediction. However, when N is larger, the copying-based method is inefficient. Eq. (4.6) shows that when a sample has large index (far away from x_0), the prediction weight should be small. To illustrate the coding gain difference between the optimal prediction weights and the copying-based method when N changes, one example is given as follows: a 1-D first-order Gaussian Markov chain [x_0 x_1 x_2 … x_N] with $\rho = 0.95$. For different N from 4 to 32, the coding gains with optimal prediction weights derived in Eq. (4.6) and the copying-based intra prediction are shown in Figure 4.1. It is clear that the copying-based prediction is always less efficient than the optimal prediction. Moreover, the coding gain difference between these methods is larger when N is increasing.

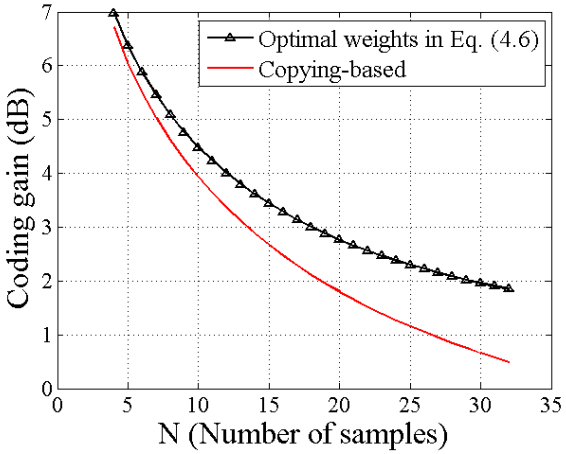


Figure 4.1. Coding comparison under different sizes of samples. For larger size samples, the copying-based method is much less efficient than the optimal prediction weights.

4.2.2 *Intra Prediction for 1-D Source with Error*

In the previous 1-D case, it shows that when the predicted samples are far away from the reference samples, the copying-based intra prediction is not a good approximation of the optimal intra prediction. Note that, in this case, the reference sample has no error. However, in a real sequence, the reference sample is always a reconstructed one, with quantization error. Hence, the reference sample is

$$\hat{x}_0 = x_0 + \delta, \tag{4.11}$$

where δ is the error on the reference pixels which has zero-mean, independent to all other samples, and has a variance of σ^2. It has been pointed out in [61] that δ can cover the case that the reference sample is unavailable (in practical video coding, the unavailable reference pixels are always filled with neighboring known pixels), or when there is noise on the original reference samples.

With the errors in the reference samples, the sequence and its auto-correlation matrix is rewritten as

$$\hat{X} = \begin{bmatrix} \hat{x}_0 & x_1 & \ldots & x_N \end{bmatrix}^T, \text{ and} \tag{4.12}$$

$$E[\hat{X}\hat{X}^T] = \begin{pmatrix} 1+\sigma^2 & \rho & \rho^2 & \cdots & \rho^N \\ \rho & 1 & \rho & \cdots & \rho^{N-1} \\ \rho^2 & \rho & 1 & \cdots & \rho^{N-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho^N & \rho^{N-1} & \rho^{N-2} & \cdots & 1 \end{pmatrix}. \tag{4.13}$$

Note that the only difference between Eq. (4.5) and Eq. (4.13) is the auto-correlation of the reference sample. The optimum prediction weights in the prediction matrix P of Eq. (4.2) are:

$$p_i = \rho^i / (1 + \sigma^2),\ i = 1, \ldots, N. \tag{4.14}$$

Comparing Eq. (4.14) with Eq. (4.6), it shows that smaller intra prediction weights are desired in the case when the reference samples have error. For a relatively large error δ, the optimum weights derived by Eq. (4.6) is sub-optimum. To illustrate the effects of this reference deviation, one example is shown as follow. For a source with $\rho = 0.95$, $N = 8$, and $\sigma = 0 \sim 0.5$, the coding gain with three intra prediction methods: 1) derived by Eq. (4.14), 2) derived by Eq. (4.6), and 3) copying-based prediction are shown in Figure 4.2. The figure shows that when the deviation associated with the reference samples increases, the intra prediction weights in Eq. (4.14) considering the error gives much better coding gain than weights in Eq. (4.6) without considering the deviation and the copying-based method.

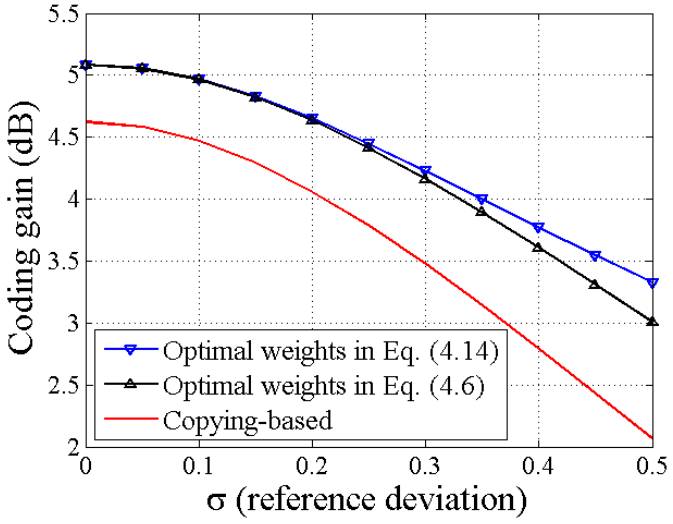


Figure 4.2. Coding gain comparison under different levels of reference deviation.

4.2.3 *Intra Prediction for 2-D Directional Sources without Error*

The previous analysis on the 1-D case can be extended to the 2D case. Denote $x_{i,j}$, (i, j = 0, …, N) as the pixel on the i-th row and j-th column in an image. Under the 2-D first-order Gaussian Markov model with zero mean and unit variance, an isophote auto-correlation model of two pixels $x_{i,j}$ and $x_{p,q}$ is used as follows:

$$E[x_{i,j}x_{p,q}] = \rho^{\sqrt{|i-p|^2+|j-q|^2}}, \tag{4.15}$$

where ρ is the correlation coefficient.

Considering the image source has a dominating direction, the correlation model can be extended to [70]

$$E[x_{i,j}x_{p,q}] = \rho^{\sqrt{d_1^2(\alpha)+\eta^2 d_2^2(\alpha)}}, \tag{4.16}$$

where,

$$\begin{cases} d_1(\alpha) = (j-q)\cos\alpha - (i-p)\sin\alpha \\ d_2(\alpha) = (j-q)\sin\alpha + (i-p)\cos\alpha \end{cases}. \tag{4.17}$$

In this model, α is the dominating direction, e.g., α = 0 means the source has a vertical direction. d_1 and d_2 are the distance between $x_{i,j}$ and $x_{p,q}$ along the dominating direction and perpendicular to that direction, respectively. η (≥ 1) is the parameter indicating the directionality. A larger η means the correlation along the dominating direction is higher than the perpendicular direction. A typical set of parameters ρ = 0.99 and η = 5 (1 for DC and Planar modes) is used. These parameters are trained on the real video sequences and the details are described in the Appendix A.

For a 2-D source, the coding block and its neighboring reference pixels can be concatenated into a tall vector. As in H.265/HEVC, for an $N\times N$ block, the $2\times N$ pixels above and above-right, and $2\times N$ pixels left and left-bottom to the block, as well as the top-left pixel, are used as the reference pixels, as shown in Figure 4.3.

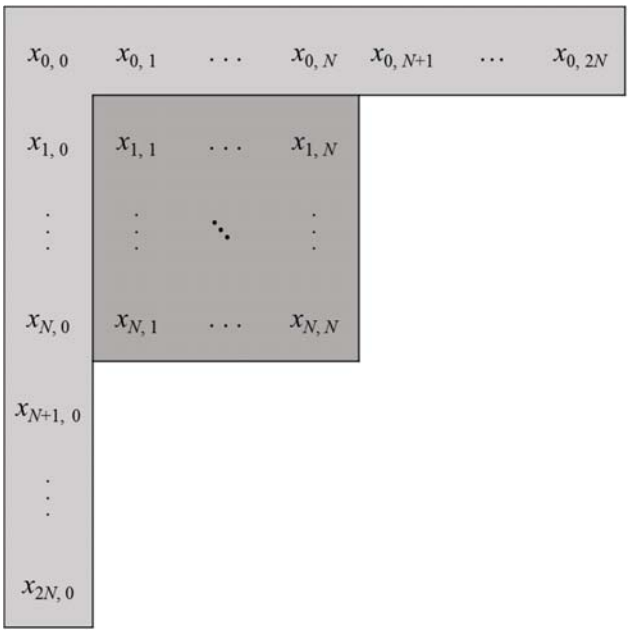


Figure 4.3. Current coding block and its neighboring reference pixels.

Here it assumes all reference pixels are available (or filled with some methods). So there is a total of $4\times N+1+N^2$ samples in the vector:

$$X = [x_{0,0}, x_{1,0}, \ldots, x_{2N,0}, x_{0,1}, \ldots, x_{0,2N}, x_{1,1}, x_{2,1}, \ldots, x_{N,N}]^T . \tag{4.18}$$

The first $4\times N+1$ samples are the references, followed by N^2 samples in the current coding block, which is concatenated column-wise. For simplicity, the 1-D vector X is rewritten as

$$X = [A^T B^T]^T , \tag{4.19}$$

where $A = \{a_i\}$, $i = 1, \ldots, 4N + 1$, are the reference samples and $B = \{b_j\}$, $j = 1, \ldots, N^2$ are the coding block samples.

The intra prediction can also be formulated as a matrix multiplication, similar to Eq. (4.2) and Eq. (4.3). The prediction matrix P is

$$P = \begin{pmatrix} \mathbf{0}_{(4N+1)\times(4N+1)} & \mathbf{0}_{(4N+1)\times N^2} \\ W_{N^2\times(4N+1)} & \mathbf{0}_{N^2\times N^2} \end{pmatrix}, \tag{4.20}$$

where $W = \{w_{i,j}\}$ ($i = 1, \ldots, N^2, j = 1, \ldots, 4N+1$) is the matrix of weights. Each row of matrix is a set of prediction weights on all reference pixels.

Similar to the 1-D source, the residual is derived by Eq. (4.7) and the coding gain can be calculated via Eq. (4.10). The problem of finding the optimum prediction weights is formulated as

$$W_{\text{opt}} = \arg\min_{W} E\left[\left\| B - W \bullet A \right\|_2^2\right]. \tag{4.21}$$

Let w_i be the i-th row of W, the problem in Eq. (4.21) is equivalent to solving each row of W,

$$w_{i,\text{opt}} = \arg\min_{w_i} E\left[\left\| b_i - w_i \bullet A \right\|_2^2\right], i = 1,..,2N . \tag{4.22}$$

The solution is

$$w_{i,\text{opt}} = E(AA^T)^{-1} \bullet E(b_i A), \tag{4.23}$$

where $E(AA^T)$ is the auto-correlation matrix of reference samples and $E(b_iA)$ is the correlation between the sample b_i and all reference samples. Actually, Eq. (4.22) is the Weiner filter.

Similar to the 1-D case, for a large block, the pixels far away from the reference boundary has lower correlation. One example as follow shows the coding gain difference between the copying-based intra prediction and the optimal weights intra prediction. In this example, the 2-D samples follow the model in Eq. (4.16) with $\rho = 0.99$, $\eta = 5$. Two directions $\alpha = 0$ and $\alpha = -\pi/4$ are tested, which means the block has a horizontal direction or a diagonal down right direction. For $\alpha = 0$, the copying-based method follows the intra prediction mode 10, and for $\alpha = -\pi/4$, the copying-based method follows the intra prediction mode 18. The block size varies from 4 to 32. The coding difference under these two directions are shown in Figure 4.4.

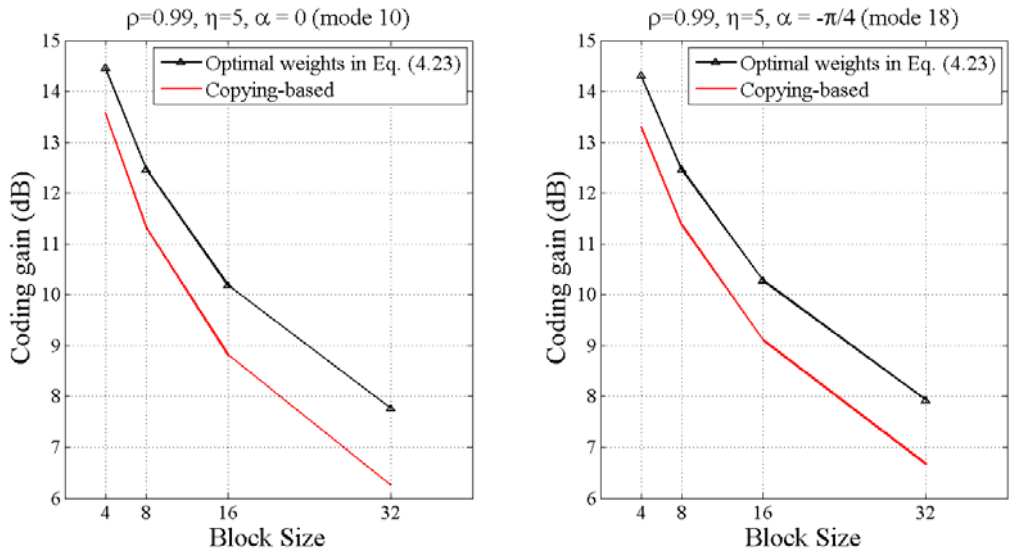


Figure 4.4. Coding gain comparison between optimal weights intra prediction and the copying-based method under two directional models (left: horizontal, right: diagonal-right-down).

From above analysis, it is similar to the 1-D case that in a large block, when there are more samples far away from the reference samples, the copying-based prediction is less efficient.

4.2.4 *Intra Prediction on 2-D Directional Sources with Error*

It has been shown in Section 4.2.2 that in the 1-D source, the error on reference samples affects the optimum intra prediction weights. The effects on 2-D sources are analyzed as follows.

Denote the distorted reference samples as

$$\overline{x}_{i,j} = x_{i,j} + \delta_{i,j}, \tag{4.24}$$

where $\delta_{i,j}$ is zero-mean, independent to all the samples and has a variance of σ^2. Hence, the Eq. (4.19) becomes

$$X = \begin{bmatrix} \overline{A}^T & B^T \end{bmatrix}^T, \tag{4.25}$$

where $\overline{A}$ is the vector of deviated reference samples. The solution in Eq. (4.22) is updated with

$$w_{i,\text{opt}} = E(\overline{A}\overline{A}^T)^{-1} \bullet E(b_i \overline{A}). \tag{4.26}$$

Comparing Eq. (4.26) to Eq. (4.23), note that $\delta_{i,j}$ is independent to b_i, the change of optimal prediction weights only comes from the inverse of new auto-correlation matrix. The effects of reference deviation on the coding gain are shown in Figure 4.5. In these figures, two different directional models (horizontal and diagonal-down-right) are tested. The σ is from 0 to 0.5. It can be seen that when the deviation is relatively large, the optimal weights in Eq. (4.26) considering the deviation is much better than the weights by Eq. (4.23) which does not consider the deviation.

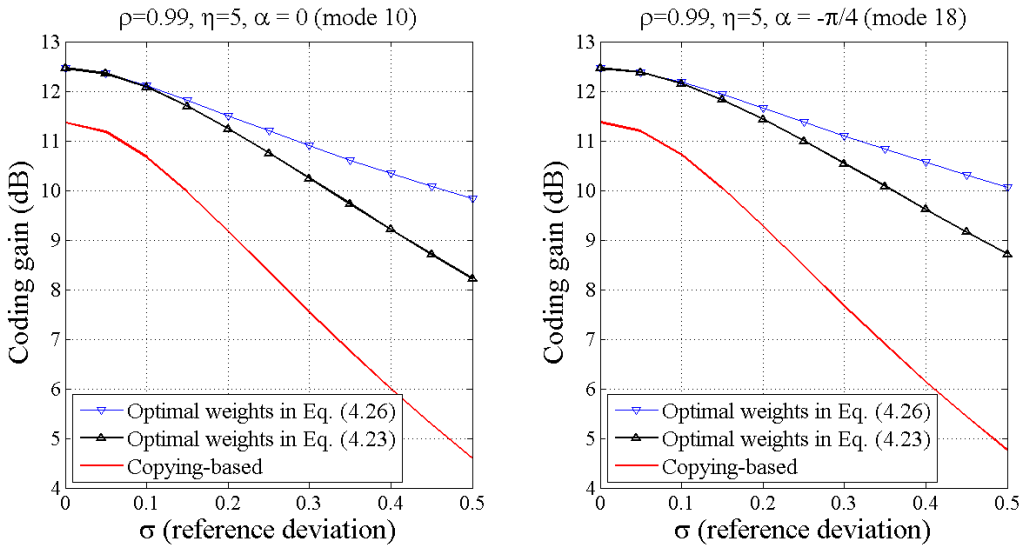


Figure 4.5. Coding gain comparison under different levels of reference deviations. When the deviation is large, the optimal weights derived in Eq. (4.26) considering the deviation is much better than the weights derived in Eq. (4.23) without considering the deviation.

From the previous analysis, we have the conclusion that the current copying-based method in intra prediction is less efficient in two cases: 1) when more predicted samples are far away from the reference samples, and 2) the reference samples have relatively larger deviations. One straight-forward idea is to apply the optimal weights instead of the copying-based method in these two cases. Since all the correlation can be derived via the 2-D source model in Eq. (4.16), or by training real image data [58], an optimum prediction weights table can be calculated. However, the weights are position-dependent, the size of weights table would be $N^2 \times (4N+1)$. In H.265/HEVC, the N could be 4, 8, 16 and 32, hence there are 151120 numbers in a weights table for one direction. Consider there are 34 intra prediction directions, each one having a different correlation matrix, there are more than 4×10^6 numbers! Hence, the position-dependent weights are not feasible for the H.265/HEVC intra prediction scheme. Hence, we like to find an approximation to the optimal weights.

From Eq. (4.6) and Eq. (4.14) derived in the 1-D case, it is clear that smaller weights would be better for the aforementioned two cases where the copying-based method is inefficient. However, the derivation of the analytical solution to the optimum weights in 2-D case is too complicated. A quantitative simulation is shown to compare the weights. In this simulation, a 4×4 block is predicted by its 17 neighboring references samples as shown in Figure 4.3. Each sample is zero-mean and unit-variance. The correlation among all 33 samples is modeled by Eq. (4.16) with $\rho = 0.99$, $\eta = 5$ and $\alpha = 0$, which is the horizontal direction. For all samples in the block, the optimal weights can be derived, given the error δ (with variance σ). Different $\sigma = 0$, 0.2 and 0.5 are simulated and the corresponding weights are shown in Figure 4.6.

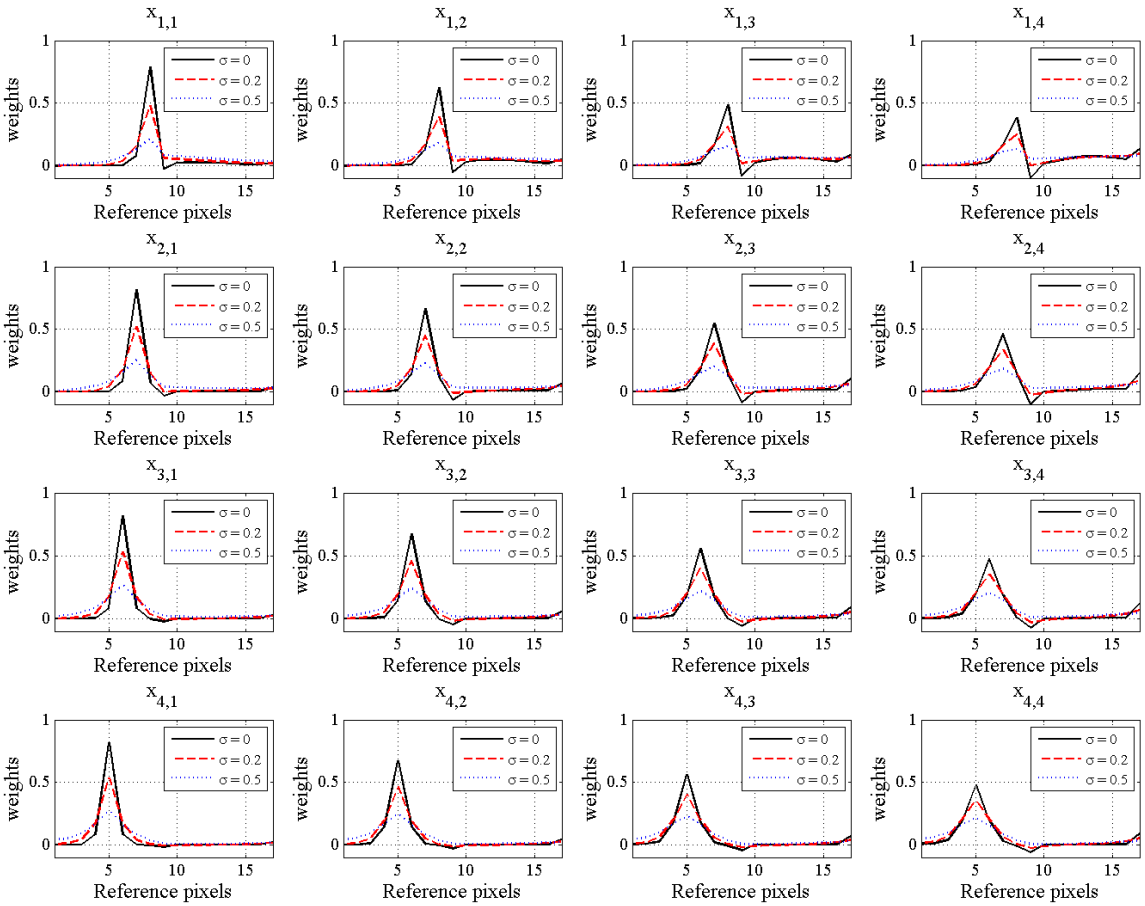


Figure 4.6. These 16 figures show the prediction weights on all 4×4 pixels. In each figure, three lines present the weights of each reference pixels under different error level (solid line: σ = 0, dashed line: σ = 0.2 and dotted line: σ = 0.5). The index of the reference pixels correspond to the position of 17 reference pixels in the clock-wise direction, for the bottom-left $x_{8,0}$ to the top-right $x_{0,8}$.

The following observations can be derived from Figure 4.6.

1) Under the same level of deviation, the prediction weights on the pixels far away from the reference along the prediction direction are smaller. In this example, the prediction direction is horizontal. From left to the right, the optimal weights under the same deviation level are decreasing.
2) On the same pixel, when the deviation level increases, the weights are also decreased. It can be proved that when the error is very large, the weights will approach to the same value, which is shown in the Appendix B.

Note that in the above example, the pixels are assumed to be zero-mean, so that the sum of weights can be less than 1, which does not cause DC drift. However, in the real image, the pixel value is among 0 to 255, the DC value should be subtracted from the reference samples so that they are zero-mean, and then a weighted prediction is applied on these samples and the DC value is added to keep the samples from DC drifting. If the DC value is estimated by using the average of neighboring pixels, it can be seen that a smaller prediction weight for zero-mean sources leads to a smoother weighted prediction on neighboring references. In the next section, we propose to use an iterative filtering method to approximate the smoothing weighted prediction.

4.3 PROPOSED RATE-DISTORTION OPTIMIZED ITERATIVE FILTERING INTRA PREDICTION

The example in the previous section shows that smoothed prediction weights instead of copying-based method is more suitable for pixels far away from the reference samples along the direction of prediction and when the reference samples have deviation. This section proposes an iterative filtering method as an extra intra prediction mode in addition to the copying-based method to handle this situation.

4.3.1 *Iterative Filtering Intra Prediction*

To smooth the prediction weights, a copying-based predictor block is first generated, then the predictor block is convolved with a smoothing filter. Since the larger block has more pixels far away from the boundary, a stronger smoothing is needed. To have a smoother predictor, a larger smoothing filter is needed or the smoothing filter is applied multiple times. In this scheme, the latter one is chosen since it is easier to choose the filtering times (discussed later) than to design the filter parameters in a large filter. The iterative filtering algorithm is summarized as follows:

Algorithm Iterative Filtering Intra Prediction

Input: An $N \times N$ coding block with its neighboring $4N+1$ reference pixels, smoothing filter K and filtering iteration numbers T.

Output: An $N \times N$ prediction block *Pred*

1. Generate an intra predictor block Blk_0 using the standard copying-based H.265/HEVC angular intra prediction method as an initial block.
2. Convolve the Blk_0 with filter kernel K recursively T times.

for $i = 1 : T$ **do**

Blk_i, = conv(Blk_{i-1}, K)

end for .

The conv() function means the Blk_{i-1} is convolved with kernel K. Note that although the left and upper reference boundary pixels are used in the convolution to determine the center pixel value, their values are not changed.

3. $Pred = Blk_T$

4.3.2 *Filter Design and Iteration Number Selection*

Given a smoothing filter, the proper iteration time needs to be determined. In this part, the coding gain defined in Eq. (4.9) is used to evaluate the best iteration number. According to our simulation, the results are not sensitive to the selection of filter kernel. It is pointed out in [60] that a mode dependent kernel can be used. However, according to our tests, mode-dependent kernels do not improve the performance of our method. It is also possible to apply multiple filters and select the best one by R-D optimization, but the extra overhead to indicate the kernel will increase the bitrate and the computing complexity. Hence, the following simple smoothing filter is used:

$$K_1 = \frac{1}{6}\begin{pmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{pmatrix} \tag{4.27}$$

Consider an $N \times N$ block with $4N+1$ neighboring references pixels where the pixel values are zero-mean and unit-variance. The correlation among all pixels is modeled by Eq. (4.16) with ρ = 0.99, $\eta = 5$ and $\alpha = 0$, which is the horizontal direction. The deviation with $\sigma = 0.2$ is added on the reference pixels. The following 3 intra prediction schemes are compared: 1) Optimal intra

prediction derived by Eq. (4.26), 2) Iterative filtering intra prediction via the filter K_1 in Eq. (4.27) with iteration time of $T = 1, 2, \ldots, 20$, and 3) the copying-based intra prediction in H.265/HEVC with Mode 10 (horizontal); The coding gain comparison under different T is showed in Figure 4.7.

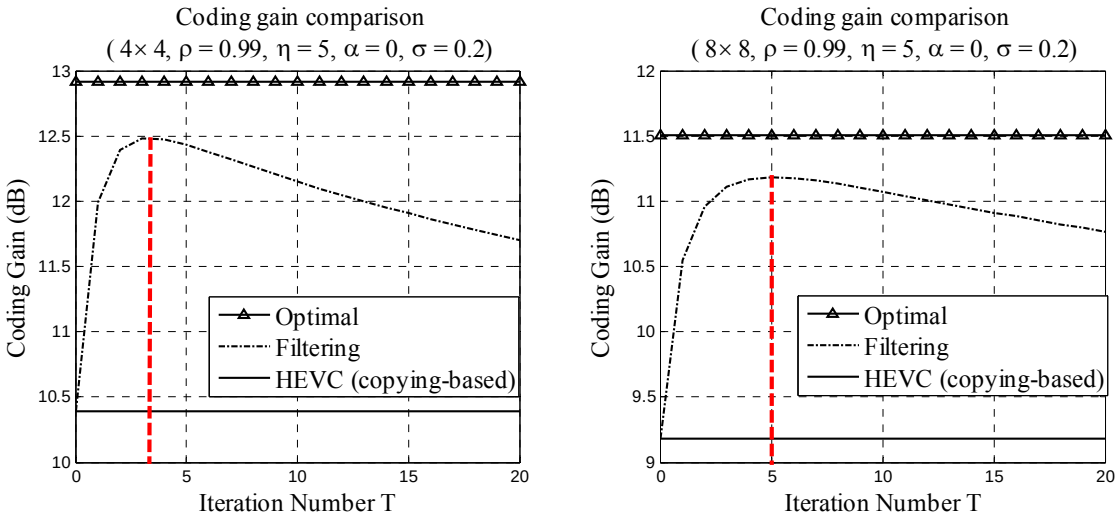


Figure 4.7. Coding comparison between 1) the optimal intra prediction; 2) the iterative filtered intra prediction and 3) the H.265/HEVC (copying-based) intra prediction mode 10. The best performance of iterative filtering intra prediction for 4×4 and 8×8 blocks appear at iteration time $T = 3$ and 5, respectively.

From the above example, it shows that the optimal weights scheme outperforms both copying-based and the proposed iterative filtering intra prediction schemes. However, as mentioned in the previous section, the optimal weights intra prediction needs a large weights table, which is infeasible in the current H.265/HEVC standard. For the iterative filtering scheme on 4×4 and 8×8 blocks, the coding gain increases as the iteration grows and then decreases after the best iteration number. The best iteration number of 3 and 5 achieve the highest gain in 4×4 and 8×8 blocks, respectively. Hence, in the design of the proposed intra prediction, the iteration number in the horizontal mode is set to 3 for 4×4 and 5 for 8×8.

The similar simulation is done on the DC mode and other H.265/HEVC angular modes, with different size include 4×4, 8×8, 16×16 and 32×32. The simulation results for the DC mode is shown in Figure 4.8.

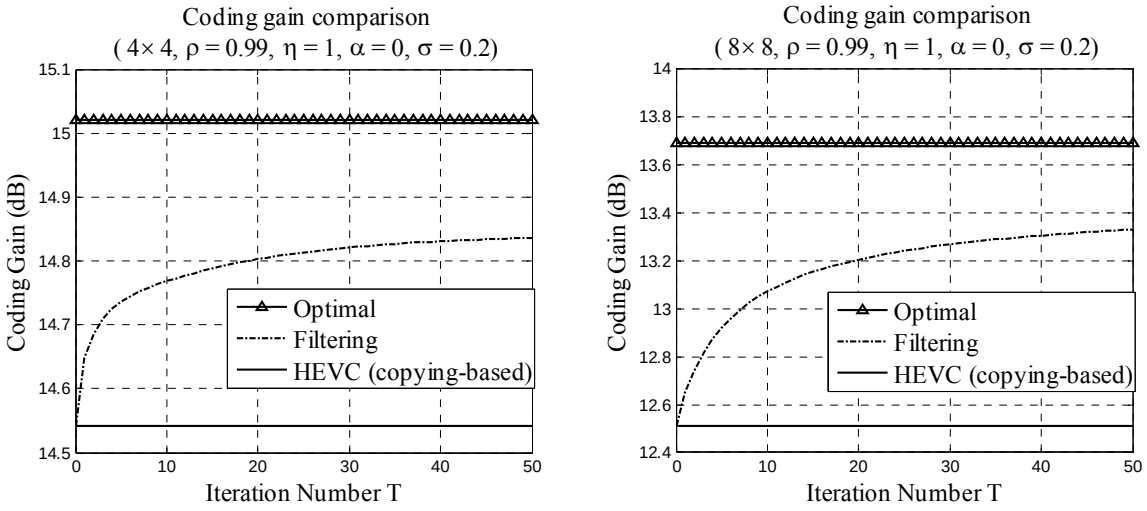


Figure 4.8. Coding comparison between the optimal, H.265/HEVC Mode 1 (DC), and the iterative filtered intra prediction schemes on a 2-D source without dominating direction ($\eta = 1$, $\rho = 0.99$ and $\sigma = 0.2$). The coding of iterative filtering intra prediction converges after about $T = 30$.

The best iteration numbers under different situations are listed in Table 4.1.

Table 4.1. Best Iteration Numbers under Different Modes and Block Sizes.

Modes	4×4	8×8	16×16	32×32
0 and 1 (Planar and DC)	32	32	32	32
2 ~ 34 (Angular Modes)	3	5	10	25

4.3.3 *Rate-Distortion-Based Mode Selection*

The proposed iterative filtering method is used as an extra mode to the existing copying-based

method in the intra prediction process. The encoder and decoder flow-chart is shown in Figure 4.9.

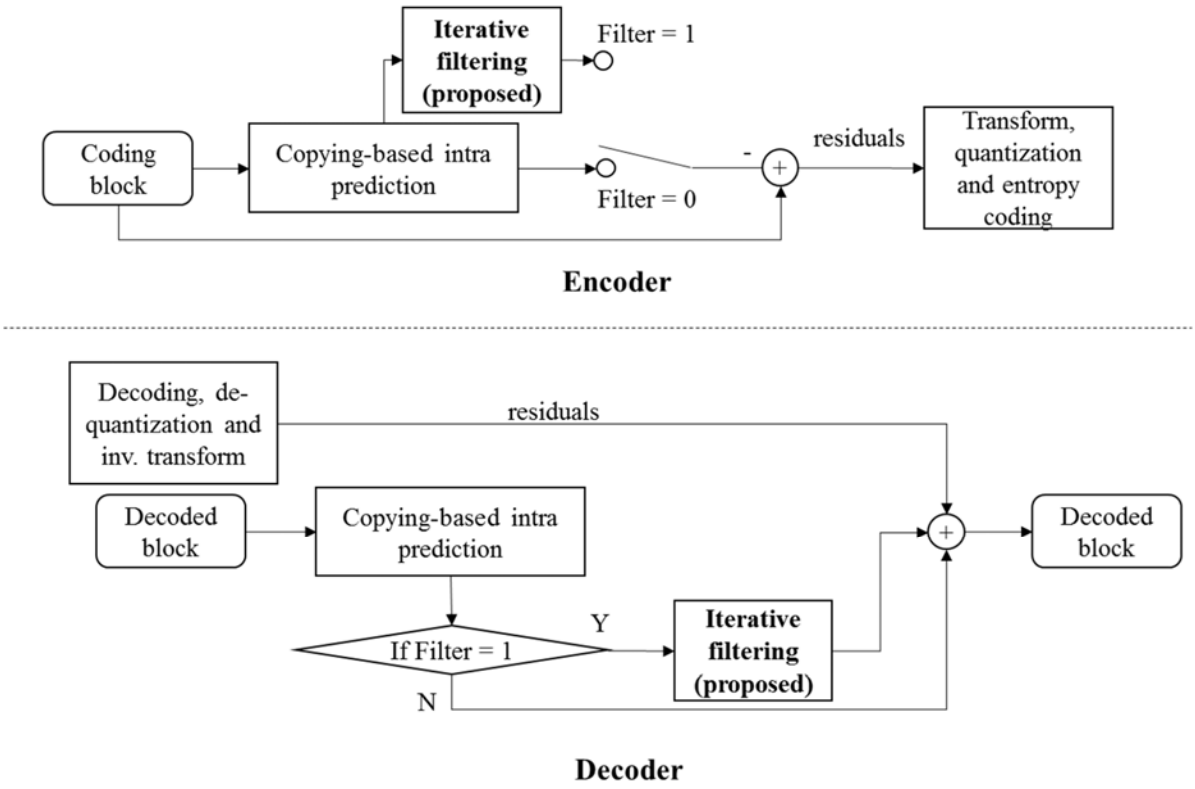


Figure 4.9. The flowchart of the proposed intra prediction scheme on both encoder and decoder side. The iterative filtering intra prediction is added as an extra coding mode.

The Filter flag in Figure 4.9 is selected based on the rate-distortion cost of each Coding Unit (CU) and encoded into the bit stream by the entropy encoder. Note that in the current H.265/HEVC standard, the intra prediction is applied on the Transform Unit (TU) level, which means the CU can be further split into a quad-tree structure and then the intra prediction is applied on the leaves of this quad-tree, which is called residual quad-tree (RQT). RQT can enhance the prediction accuracy since the reference is closer to the predicted block. To achieve the best performance, the proposed iterative filtering should be tested on each TU and different TUs may choose different

modes (with or without filtering). However, this increases overhead bits and the searching time significantly. Therefore, in the proposed scheme, the RDO-based selection is tested on the CU level and only one Filter flag is encoded for each CU.

4.4 EXPERIMENTAL RESULTS

4.4.1 *Bitrate Reduction in H.265/HEVC*

The proposed intra prediction method is implemented in the H.265/HEVC reference software, HM-14.0 [51], on all Y, Cb, and Cr components. The coding parameter configuration follows the H.265/HEVC common test conditions [71]. The main tier (QP = 22, 27, 32 and 37) is tested. Since we are more interested in high resolution video sequences, 2K to 8K sequences are tested. In this test, the kernel in Eq. (4.27) and the iteration numbers in Table 4.1 are used.

Since better intra-coded frames can also improve the coding efficiency of inter coding. Besides all intra (AI) coding, we also test our proposed method in random access (RA), low-delay B (LDB), and low-delay P (LDP) coding configures. Note that in these inter coding tests, all frames are coded. The BD-Bitrate [41] results for Y components are shown in Table 4.2. The negative numbers mean the bitrate is reduced compared with HM-14.0.

Table 4.2. BD-bitrate Reduction with AI, RA, LDB and LDP Configuration on Full Sequences ("-" Means Gain).

		AI	RA	LDB	LDP
2K (1920×1080)	Kimono	-2.7%	-1.0%	-0.2%	-0.7%
	ParkScene	-1.6%	-1.0%	-0.5%	-0.6%
	Cactus	-1.5%	-0.8%	-0.4%	-0.7%
	BasketballDrive	-1.7%	-1.0%	-0.5%	-1.0%
	BQTerrace	-0.6%	-0.3%	-0.2%	-0.1%
4K	Chuno_s4	-2.3%	-1.3%	-0.9%	-1.4%
	Chuno_s31	-3.2%	-2.0%	-2.1%	-2.1%
	Crowdrun	-2.2%	-1.2%	-0.5%	-0.8%

(3840×2160, except hotel 4096×2048)	Hotel	-1.8%	-1.4%	-1.0%	-1.2%
	Pku_girl	-2.4%	-1.7%	-1.1%	-1.4%
	Reed	-2.1%	-1.5%	-1.3%	-1.6%
	4k_seq1	-3.7%	-2.7%	-2.0%	-2.2%
	Butterfly	-3.2%	-2.2%	-1.1%	-2.1%
8K (7680×4320)	8K_seq7	-2.0%	-0.8%	-0.3%	-0.7%
	8K_seq12	-3.0%	-1.1%	-0.6%	-1.1%
	8K_seq14	-3.8%	-2.9%	-1.4%	-2.3%
	8K_seq17	-2.6%	-1.2%	-0.6%	-0.7%
	8K_seq18	-1.8%	-1.3%	-1.0%	-1.2%
	DS_store	-0.9%	-0.7%	-0.7%	-0.7%
	2K average	-1.6%	-0.8%	-0.4%	-0.6%
	4K average	-2.6%	-1.8%	-1.2%	-1.6%
	8K average	-2.4%	-1.3%	-0.7%	-1.1%
	All average	-2.3%	-1.4%	-0.9%	-1.2%

The gain of this proposed method is 2.3% on average and up to 3.8% in all intra (AI) coding, which is significant compared to the current H.265/HEVC coding standard. It is also clear that our proposed method can also improve the coding gain in inter coding settings.

4.4.2 *Comparison with other methods*

We compare following four methods as follows:

1) Recursive extrapolation intra prediction in [53];
2) PDE-based inpainting intra prediction in [60];
3) Our proposed method with following filtering kernel and the associated iteration numbers in Table 4.3.

$$K_2 = \frac{1}{9}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}; \tag{4.28}$$

4) Our proposed method with filtering kernel K_1 as in Eq. (4.27) and iteration numbers in Table 4.1.

Table 4.3. Best Iteration Numbers under Different Modes and Block Sizes.

Modes	4×4	8×8	16×16	32×32
0 and 1 (Planar and DC)	32	32	32	32
2 ~ 34 (Angular Modes)	2	3	5	20

Note that, different from the testing in Section IV-A, we test more sequences with lower resolution (LR). Since our proposed method is an intra coding tool and the coding performance on first few frames are very good approximation of the overall performance, we only test short sequences with first 10 frames for LR and 2K sequences and first 3 frames for 4K and 8K. All results are tested with the AI configuration. The BD-Bitrate results are listed in Table 4.4.

Table 4.4. BD-bitrate Reduction Compared with Other Methods in the AI Configuration on Short Sequences ("-" Mean Gain).

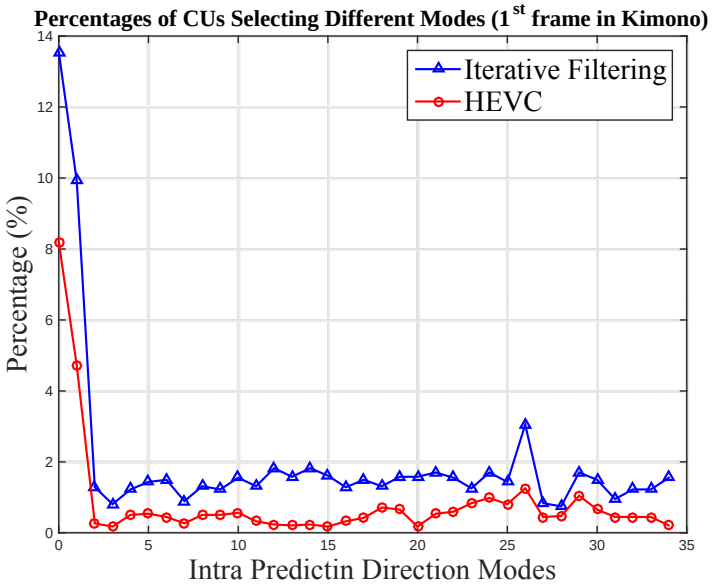
		[53]	[60]	**Ours with K_2**	**Ours with K_1**
Lower Resolution (416×240, 832×480)	BasketbalDrill	-0.3%	-0.2%	**-0.5%**	**-0.5%**
	BasketballPass	-0.4%	0.0%	**-0.2%**	**-0.4%**
	BlowingBubbles	-0.4%	-0.2%	**-0.4%**	**-0.6%**
	PartyScene	-0.4%	-0.2%	**-0.4%**	**-0.5%**
	RaceHorse	-0.7%	-0.4%	**-1.0%**	**-1.0%**
	RaceHorseC	-0.8%	-0.6%	**-1.1%**	**-1.3%**
2K (1920×1080)	Kimono	-1.2%	-0.9%	**-2.2%**	**-2.6%**
	ParkScene	-1.0%	-0.9%	**-1.5%**	**-1.7%**
	Cactus	-0.9%	-0.8%	**-1.3%**	**-1.6%**
	BasketballDrive	-2.2%	-1.0%	**-1.7%**	**-1.8%**
	BQTerrace	-0.5%	-0.3%	**-0.5%**	**-0.6%**
4K (3840×2160, except hotel 4096×2048)	Chuno_s4	-0.8%	-1.2%	**-2.0%**	**-2.1%**
	Chuno_s31	-1.8%	-0.8%	**-3.1%**	**-3.3%**
	Crowdrun	-1.3%	-1.1%	**-2.0%**	**-2.3%**
	Hotel	-1.5%	-1.2%	**-1.8%**	**-1.8%**
	Pku_girl	-1.7%	-1.5%	**-2.2%**	**-2.4%**
	Reed	-1.6%	-1.7%	**-1.8%**	**-1.9%**
	4k_seq1	-2.8%	-2.2%	**-3.3%**	**-3.6%**
	Butterfly	-1.5%	-2.1%	**-2.6%**	**-2.8%**
8K (7680×4320)	8K_seq7	-1.0%	-1.3%	**-2.0%**	**-1.9%**
	8K_seq12	-1.7%	-1.6%	**-2.7%**	**-2.9%**
	8K_seq14	-2.2%	-1.5%	**-4.2%**	**-4.0%**

	8K_seq17	-1.7%	-1.5%	**-2.3%**	**-2.5%**
	8K_seq18	-0.6%	-1.2%	**-1.8%**	**-1.8%**
	DS_store	-1.1%	-0.4%	**-0.7%**	**-0.8%**
	Encoding Time	225%	390%	**395%**	**446%**
	Decoding Time	105%	122%	**143%**	**186%**
	LR average	-0.5%	-0.3%	**-0.6%**	**-0.7%**
	2K average	-1.2%	-0.8%	**-1.6%**	**-1.6%**
	4K average	-1.5%	-1.5%	**-2.4%**	**-2.4%**
	8K average	-1.4%	-1.3%	**-2.3%**	**-2.3%**
	All average	-1.2%	-1.0%	**-1.7%**	**-1.9%**

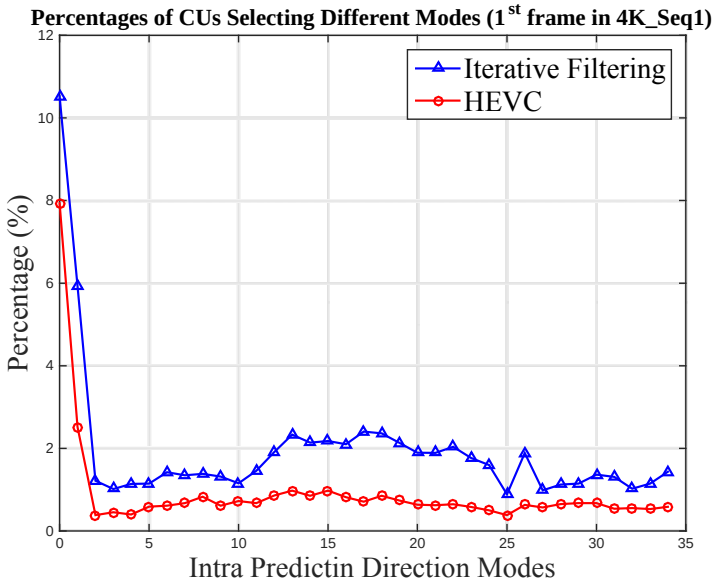
Compared with other methods in [53] and [60], the proposed method outperforms those two on almost all testing sequences. Comparing the results from kernel K_1 and kernel K_2, it can be seen that the gain is not sensitive to the selection of kernel, as long the proper iteration numbers are selected.

4.4.3 *Hit Ratio*

To demonstrate the effectiveness of the proposed iterative filtering method, the hit ratio of this new intra prediction mode is presented in Figure 4.10 and Figure 4.11. Figure 4.10 shows the percentage of CUs using difference modes in sequences “Kimono” and “4K_Seq1”. It is clear that more CUs select iterative filtering modes than the copying-based HEVC mode. From Figure 4.11, it can be seen that more than half of the CUs select the iterative filtering mode.



(a) 1st frame of Kimono



(b) 1st frame of 4K_Seq1

Figure 4.10. Percentage of CUs using iterative filtering and H.265/HEVC mode under different intra prediction directions in (a) Kimono and (b) 4K_Seq1.

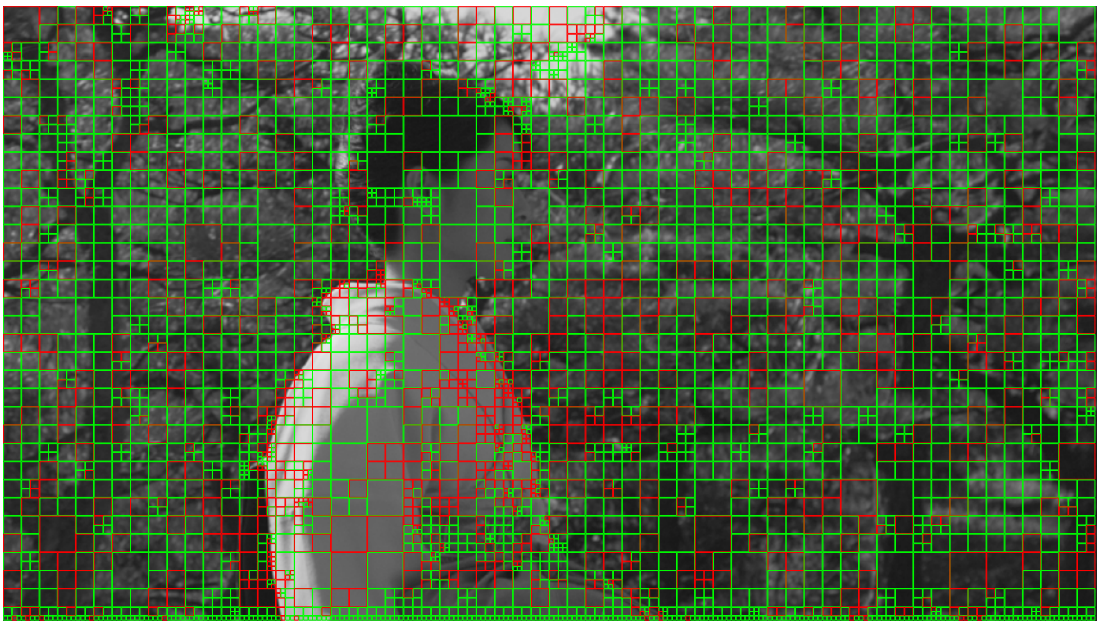


Figure 4.11. Hit ratio in the first frame of Kimono. The green blocks indicate CUs using iterative filtering mode. The red blocks indicate CUs using H.265/HEVC intra prediction mode. (Best view in color).

4.4.4 *Gain from DC, Planar and Angular modes*

To investigate the gain from different modes, we test more cases with the proposed method applied on 1) only DC mode, 2) only Planar mode, and 3) only Angular modes. The filtering kernel follows Eq. (4.27) and the iteration number is in Table 4.1. The results are shown in Table 4.5.

Table 4.5. BD-Bitrate Reduction on Separate Modes in AI Configuration on Short Sequences ("-" Means Gain).

		Only DC	Only Planar	Only Angular
Lower Resolution (416×240, 832×480)	BasketbalDrill	0.0%	0.1%	-0.6%
	BasketballPass	0.0%	0.1%	-0.4%
	BlowingBubbles	0.0%	0.0%	-0.6%
	PartyScene	-0.1%	-0.1%	-0.4%
	RaceHorse	-0.3%	-0.2%	-1.1%
	RaceHorseC	-0.4%	-0.4%	-1.2%

2K (1920×1080)	Kimono	-0.8%	-0.9%	-2.4%
	ParkScene	-1.0%	-0.9%	-1.6%
	Cactus	-0.4%	-0.5%	-1.5%
	BasketballDrive	-0.3%	-0.3%	-1.8%
	BQTerrace	-0.2%	-0.2%	-0.5%
4K (3840×2160, except hotel 4096×2048)	Chuno_s4	-0.6%	-0.6%	-2.0%
	Chuno_s31	-0.9%	-1.4%	-3.0%
	Crowdrun	-0.8%	-0.8%	-2.1%
	Hotel	-0.5%	-0.4%	-1.8%
	Pku_girl	-0.8%	-0.7%	-2.3%
	Reed	-0.9%	-0.8%	-1.8%
	4k_seq1	-0.8%	-0.9%	-3.4%
	Butterfly	-0.3%	-0.3%	-2.8%
8K (7680×4320)	8K_seq7	-0.3%	-0.4%	-1.9%
	8K_seq12	-0.7%	-0.6%	-2.8%
	8K_seq14	-0.7%	-1.4%	-3.8%
	8K_seq17	-0.8%	-0.6%	-2.4%
	8K_seq18	-0.5%	-0.6%	-1.8%
	DS_store	-0.2%	-0.2%	-0.8%
	Encoding Time	237%	180%	422%
	Decoding Time	190%	156%	185%
	LR average	-0.1%	-0.1%	-0.7%
	2K average	-0.5%	-0.6%	-1.5%
	4K average	-0.7%	-0.7%	-2.4%
	8K average	-0.5%	-0.6%	-2.2%
	All average	-0.5%	-0.5%	-1.8%

The result shows that applying only on DC and only on Planar achieve the similar gain. Applying on only angular modes can achieve a gain of 1.8%, which is close to the gain on all modes. One reason that angular modes contribute most of the gain is that most blocks use angular modes. Another reason is that the smoothed angular blocks sometimes are similar to smoothed DC or smoothed Planar predicted blocks, so that some blocks originally using the DC or Planar modes would change to the angular modes. This is also the reason why the gain on separate modes are not additive.

4.4.5 *Effects of iteration numbers*

The number of convolution iterations affects the computation complexity. To investigate the coding performance of the proposed method with different iterations, more tests on the kernel as in Eq. (4.27) are conducted with the following iteration numbers (Table 4.6):

Table 4.6. Iteration Numbers for Tests with Different Iteration Numbers.

	DC and Planar	Angular (4×4, 8×8, 16×6, 32×32)
Test 1	2	1, 1, 1, 1
Test 2	4	2, 2, 2, 2
Test 3	50	10, 20, 20, 30

The BD-Bitrate results are shown in Table 4.7.

Table 4.7. BD-bitrate Reduction Compared with Different Iteration Numbers in AI Configuration on Short Sequences ("-" Means Gain).

		Test 1	Test 2	Test 3
Lower Resolution (416×240, 832×480)	BasketbalDrill	-0.3%	-0.4%	-0.4%
	BasketballPass	-0.3%	-0.3%	-0.1%
	BlowingBubbles	-0.5%	-0.5%	-0.3%
	PartyScene	-0.5%	-0.5%	-0.2%
	RaceHorse	-0.9%	-1.0%	-0.7%
	RaceHorseC	-0.9%	-1.2%	-0.9%
2K (1920×1080)	Kimono	-0.5%	-0.7%	-2.8%
	ParkScene	-1.0%	-1.4%	-1.4%
	Cactus	-1.1%	-1.2%	-1.1%
	BasketballDrive	-1.4%	-1.5%	-1.5%
	BQTerrace	-0.9%	-0.7%	-0.3%
4K (3840×2160, except hotel 4096×2048)	Chuno_s4	-1.7%	-2.0%	-1.4%
	Chuno_s31	-0.9%	-1.2%	-3.3%
	Crowdrun	-1.4%	-1.8%	-1.8%
	Hotel	-1.6%	-1.7%	-1.5%
	Pku_girl	-1.2%	-1.6%	-1.9%
	Reed	-0.7%	-1.1%	-1.8%
	4k_seq1	-1.4%	-2.1%	-3.2%
	Butterfly	-1.6%	-2.2%	-1.9%

8K (7680×4320)	8K_seq7	-1.4%	-1.7%	-1.4%
	8K_seq12	-1.8%	-2.4%	-2.1%
	8K_seq14	-2.2%	-3.0%	-3.6%
	8K_seq17	-2.0%	-2.4%	-1.8%
	8K_seq18	-1.0%	-1.2%	-1.5%
	DS_store	-1.0%	-1.0%	-0.5%
	Encoding Time	239%	262%	679%
	Decoding Time	117%	124%	256%
	LR average	-0.6%	-0.6%	-0.4%
	2K average	-1.0%	-1.1%	-1.4%
	4K average	-1.3%	-1.6%	-2.1%
	8K average	-1.6%	-1.9%	-1.8%
	All average	-1.1%	-1.3%	-1.4%

The result shows that if the iteration number is too small or too larger, the performance gain is reduced (compared to the best result of 1.9%, in Table 4.4, last column).

4.4.6 *Encoding and Decoding Complexity*

Note that this new intra prediction mode can be done in parallel to current H.265/HEVC mode. Also, in most practical applications, both inter coding and intra coding are used. Since inter coding requires much more computations than the intra coding, the overall impact on the complexity is not as high. For inter-frame coding, our experimental results show that the additional encoding time is 25% (RA), 23% (LDB) and 33%(LDP) and the additional decoding time is increased by 0%(RA), 0%(LDB), and 2%(LDP).

For the all-Intra coding, according to the encoding and decoding time in Table 4.4, the additional encoding time that the proposed new intra prediction mode induces is 295% of the anchor and the additional decoding time is 43% of the anchor. The additional complexity mainly comes from the iterative filtering process and the extra mode decision. By reducing the iteration number, the additional encoding and decoding time can be reduced. In Table 4.4 Test 2, the

additional complexity is reduce to 162% (encoding) and 24% (decoding) of the anchor.

4.5 CONCLUSION

This chapter has two contributions: 1) it discusses the intra prediction weights in the situation when the predicted samples have low correlation with the reference sample and the reference samples have deviation, and 2) proposes a novel intra prediction scheme using iterative filtering, which is suitable for cases in 1). The proposed method is used as an extra intra prediction mode in addition to the copying-based H.265/HEVC intra prediction scheme. The theoretical coding gain is used to compare the proposed scheme and the current H.265/HEVC intra prediction scheme. It shows that better gain is achieved by the iterative filtering method when a proper iteration number is selected. The experimental results in the H.265/HEVC reference software also confirms the effectiveness. The BD-bitrate reduction is up to 3.8% and 2.3% on average.

Chapter 5. FILTERING FOR MOTION COMPENSATED PREDICTION

5.1 BACKGROUND

H.265/HEVC (High Efficiency Video Coding) is the latest video coding standard, and can achieve about 50% bit-rate reduction compared with its preceding standard H.264/AVC [52]. Many advanced techniques have been included in H.265/HEVC standard, including the quad-tree coding structure [72], more efficient transform coefficient coding [73], advanced intra-picture prediction [74], inter-picture prediction [75], [76] and high throughput CABAC [77]. More tools were proposed for Range Extensions (REext), and in the current ongoing Screen Content Coding (SCC), e.g., Intra Block Copy [43], Cross-component prediction [78], and palette mode [63]. However, these tools specifically designed for REext and SCC only contribute limited gain for the 4:2:0 YUV natural content video.

Some inter-frame prediction techniques have been proposed to improve the prediction efficiency, e.g., fractional-pixel motion estimation and multiple-hypothesis [79]. It is observed that the noise in the camera-captured video affects the coding efficiency significantly. By averaging two similar blocks (derived from motion estimation), the noise can be reduced. However, since the noise pattern and intensity are different in different frames, this scheme may not be sufficient. In this chapter, we propose to apply a filter in the motion compensated prediction process so that the noise can be further reduced. Two schemes (i.e., applying the filter before, or after the fractional-pixel interpolation) are investigated in this chapter.

Low-pass filtering removes noise but usually blurs the edges, which may degrade the inter-prediction, especially when the current frame is clear and the reference frame is blurred. Bilateral

filter [16] can reduce the noise and preserve the edge. Therefore, in this chapter, we investigate the use of the bilateral filter in motion compensated prediction to improve the prediction.

The remainder of this chapter is organized as follows: Section 5.2 describes two schemes of applying the bilateral filtering. Section 5.3 presents these two schemes with adaptive filtering parameters. Section 5.4 presents the experimental results. Section 5.5 concludes this chapter.

5.2 BILATER FILTER IN MOTION COMPENSATION

In 2-D images, a Gaussian kernel bilateral filter [16] has the following format:

$$\bar{I}_{i,j} = \frac{1}{w} \sum_{(x,y)\in window} I_{x,y} \bullet G_r \bullet G_s \tag{5.1}$$

$$w = \sum_{(x,y)\in w} G_r \bullet G_s \tag{5.2}$$

$$G_r = \frac{1}{\sqrt{2\pi}\sigma_r} e^{-\frac{\left|I_{i,j}-I_{x,y}\right|^2}{2\sigma_r^2}} \tag{5.3}$$

$$G_s = \frac{1}{\sqrt{2\pi}\sigma_s} e^{-\frac{d^2}{2\sigma_s^2}} \tag{5.4}$$

where $I_{x,y}$ and $\bar{I}_{i,j}$ are the original and filtered pixel value, at position (x, y) and (i, j), respectively, and the window means the filtering support window, G_r and G_s are two Gaussian kernels, for pixel intensity difference and spatial (Euclidian) distance d, and w is the weight to normalize the filtered value.

In our proposed method, an extra coding mode which applies a bilateral filter in the motion compensated prediction is introduced in the inter-frame coding. Two schemes of this extra filter mode are described as follows.

5.2.1 *Pre-interpolation filtering*

The block diagram of this filter mode is shown in Figure 5.1. In this method, we apply the bilateral filter on the reference blocks before performing fractional-pixel motion estimation. The actual rate-distortion cost is calculated, and compared with the "no filter" mode. The better mode is selected and an extra flag is encoded in the bit-stream.

Note that this is a CU(Coding Unit)-level mode selection, and for different PUs(Prediction Units) in the same CU (2N×N, N×2N, N×N, and asymmetric partitions), the same filter mode is used. Moreover, in the bi-directional prediction, where two reference blocks from List 0 and List 1 are used in the prediction, the same filter (or "no filter") is applied on reference blocks from both lists, so that no extra bit is needed for indicating the filter type in the two reference lists.

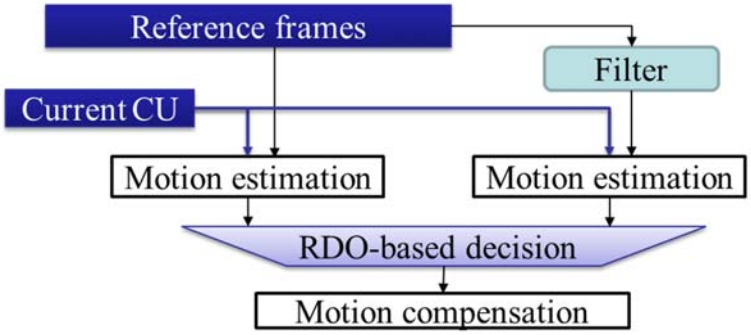


Figure 5.1. Filter mode decision for the pre-interpolation.

5.2.2 *Post-interpolation filtering*

In this scheme, the bilateral filter is applied on the filtered reference block, after the fractional-pixel motion estimation. After the interpolation, a bilateral filter is applied on the interpolated block. The block diagram of the filter mode decision for this scheme is shown in Figure 5.2. Similar

to the pre-interpolation filtering, the bilateral filter is an extra mode, and the better mode compared with the "no filter" mode is selected based on the actual R-D cost.

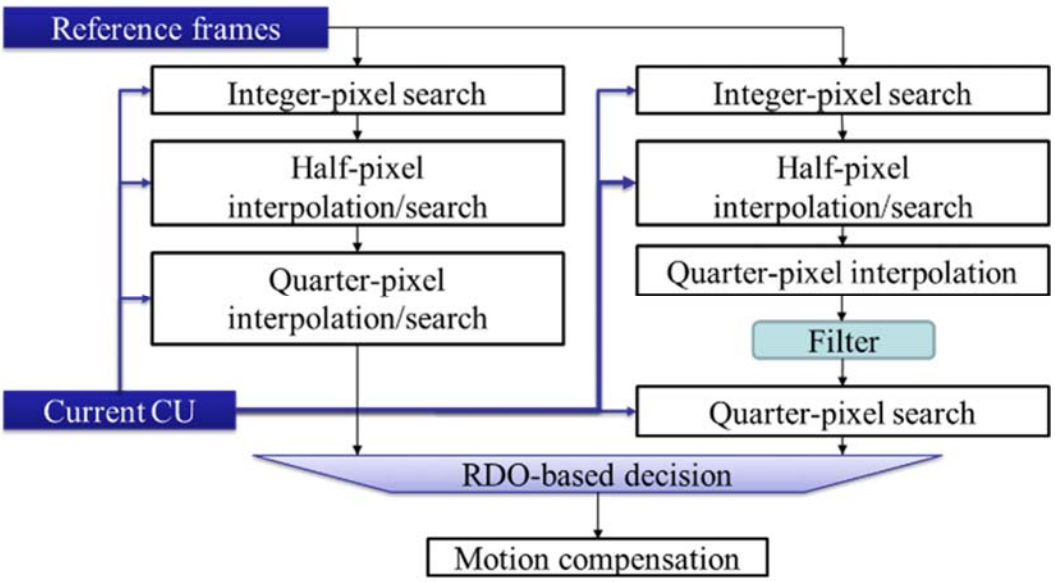


Figure 5.2. Filter mode decision for the post-interpolation.

In the experiment, the calculation of the Gaussian kernels in the bilateral filter is very time-consuming. Since the values of the distance and intensity difference are bounded, we pre-calculate the G_s and G_r with the given σ_r and σ_s and save those values in a look-up table; therefore, the calculation is performed with a look-up operation.

5.3 Bilateral Filter with Adaptive Parameters

Since different sets of filter parameters may be more suitable for different noise conditions, it is advantageous to use more than one set of filter parameters. With one extra filter mode in the motion compensated prediction (pre-interpolation or post-interpolation), the overall gains on our test sequences are about 2.6% (LDP), 1.0% (LDB), 1.1% (RA) with pre-interpolation, and 2.4%

(LDP), 0.3% (LDB), 0.4% (RA) with post-interpolation (the details of tests are described in Section IV). We could also extend these methods to even more filter parameter sets. In this chapter, we investigate an RDO-based bilateral filter, which includes the first mode as no filter, and the remaining modes as bilateral filter with different parameter sets. We implement this scheme on both pre/post-interpolation schemes. We also tested different filters besides the bilateral filter on both schemes; the results are shown in the next section.

Two issues need to be considered: 1) how to select the parameter sets, and 2) how many parameters sets are needed. Using more filters may not provide larger gain since the extra gain with a new filter would fade and the overhead to indicate the filter mode will increase. In our experiments, we tested two and three parameter sets, as well as a single filter mode discussed in Section 5.2. To determine each parameter set, we first use a large step-size to test the parameters, and once we determine whether to use smoother or sharper ones, we gradually reduce the step-size to refine the parameters. In the bilateral filter with two parameter sets, it includes the one used in the bilateral filter with one parameter set. Similarly, in the bilateral filter with three parameter sets, it includes the two previous parameter sets. The details of these parameters are shown in Section 5.4.

5.4 Experimental Results

In this section, we test our RDO-based adaptive bilateral filter in the H.265/HEVC reference software HM-14.0 [51]. The testing conditions are listed in Table 5.1. Note that we are looking at ultra HD and beyond and so we are not looking at H.265/HEVC Class D, C, and E sequences in this study. We test the pre-interpolation method with one, two, and three parameter sets, respectively. We also compare the bilateral filter with some other de-noising filter, including mean filter and median filter. The post-interpolation method is tested with only one parameter set since

its performance is less promising than the pre-interpolation method's and it requires more computational complexity. All tests we performed are listed in Table 5.2.

Table 5.1. Test Conditions.

Coding configurations	Low-delay P, Low-delay and Random Access coding
QP	22, 27, 32 and 37 (main-tier)
Test sequences	4:2:0 YUV sequences with different resolutions (2K, 4K and 8K)
Number of frames	25 to 65 depending on frame-rate. All sequences were coded for 1 second duration.

Table 5.2. Different Tests.

Test 1	Pre-interpolation filter (3×3 mean filter)
Test 2	Pre-interpolation filter (5×5 median filter)
Test 3	Pre-interpolation filter (5×5 BLF with σ_r=17.5, σ_s=1)
Test 4	Pre-interpolation filter (5×5 adaptive BLF with two parameter sets, σ_{r1}=17.5, σ_{s1}=1, σ_{r2}=54, σ_{s2}=0.85)
Test 5	Pre-interpolation filter (5×5 adaptive BLF with three parameter sets, σ_{r1}=17.5, σ_{s1}=1, σ_{r2}=54, σ_{s2}=0.85, σ_{r3}=350, σ_{s3}=0.5)
Test 6	Post-interpolation filter (9×9 BLF with σ_r=15, σ_s=2.5)

The detail results (BD-bitrate [41] change) of the pre-interpolation scheme are presented in Table 5.3, Table 5.4 and Table 5.5. The detailed results of the post-interpolation scheme are presented in Table 5.6.

Table 5.3. BD-Bitrate Difference for Pre-Interpolation Test 1 - 5 (Random Access, "-" Means Gain).

		Random Access				
		Test 1	Test 2	Test 3	Test 4	Test 5
2K (1920×1080)	Kimono	-0.6%	-0.3%	-0.6%	-0.8%	-0.9%
	ParkScene	-0.1%	-0.1%	-0.1%	-0.3%	-0.3%
	Cactus	-0.2%	0.0%	-0.4%	-0.4%	-0.6%
	BasketballDrive	-0.6%	-0.3%	-0.5%	-0.6%	-0.8%
	BQTerrace	-0.3%	-0.1%	-0.6%	-0.7%	-0.7%
4K	chuno_s4	-1.4%	-0.6%	-1.0%	-1.5%	-1.7%

(3840×2160) Hotel (4096×2048)	chuno_s31	-0.8%	-0.6%	-1.0%	-0.7%	-0.9%
	crownrun	-1.5%	-0.7%	-1.1%	-1.9%	-2.2%
	hotel	-0.3%	-0.1%	-0.4%	-0.6%	-0.5%
	Pku_girl	-0.3%	-0.2%	-0.1%	-0.4%	-0.4%
	reed	-1.3%	-0.5%	-0.7%	-1.2%	-1.3%
	4k_seq1	-0.1%	-0.1%	-0.8%	-0.8%	-0.8%
	butterfly	-8.3%	-6.2%	-6.5%	-8.3%	-8.8%
8K (7680×4320)	8K_seq7	-3.1%	-2.3%	-3.0%	-3.5%	-3.9%
	8K_seq12	-3.9%	-3.0%	-2.6%	-3.4%	-3.8%
	8K_seq14	-1.7%	-3.2%	-1.9%	-1.8%	-2.0%
	8K_seq17	-0.5%	0.1%	0.1%	-0.5%	-1.0%
	8K_seq18	-0.8%	-0.4%	-0.6%	-0.7%	-1.1%
	DS_store	-0.1%	-0.1%	-0.2%	-0.3%	-0.3%
	2K average	-0.4%	-0.2%	-0.4%	-0.6%	-0.7%
	4K average	-1.7%	-1.1%	-1.5%	-1.9%	-2.1%
	8K average	-1.7%	-1.5%	-1.4%	-1.7%	-2.0%
	All average	**-1.3%**	**-0.9%**	**-1.1%**	**-1.4%**	**-1.6%**

Table 5.4. BD-Bitrate Difference for Pre-Interpolation Test 1 - 5 (Low-Delay B, "-" Means Gain).

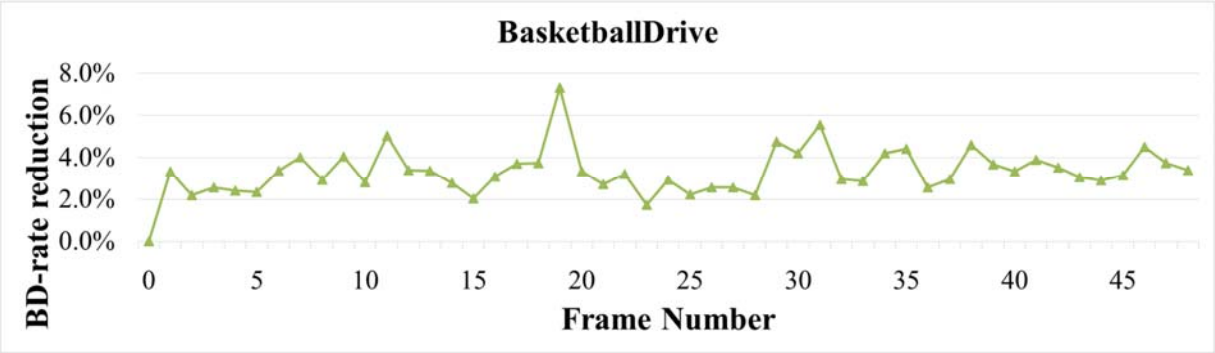
		Low delay B				
		Test 1	Test 2	Test 3	Test 4	Test 5
2K (1920×1080)	Kimono	-1.2%	-0.5%	-1.1%	-1.2%	-1.2%
	ParkScene	-0.3%	-0.5%	-0.5%	-0.6%	-0.5%
	Cactus	-0.3%	-0.3%	-0.5%	-0.6%	-0.8%
	BasketballDrive	-0.9%	-0.8%	-1.0%	-1.0%	-1.1%
	BQTerrace	-0.3%	-0.2%	-0.4%	-0.6%	-0.7%
4K (3840×2160) Hotel (4096×2048)	chuno_s4	-1.7%	-0.7%	-1.2%	-2.0%	-2.2%
	chuno_s31	-1.0%	-0.8%	-1.0%	-0.9%	-0.7%
	crownrun	-1.3%	-0.9%	-1.0%	-1.9%	-2.2%
	hotel	-0.4%	-0.3%	-0.7%	-1.0%	-0.9%
	Pku_girl	-0.4%	-0.3%	-0.4%	-0.5%	-0.6%
	reed	-1.8%	-0.5%	-0.9%	-1.8%	-1.9%
	4k_seq1	-0.2%	0.0%	-1.0%	-0.9%	-0.9%
	butterfly	-5.0%	-3.5%	-4.0%	-5.2%	-5.4%
8K (7680×4320)	8K_seq7	-2.6%	-1.6%	-2.4%	-2.8%	-3.1%
	8K_seq12	-3.1%	-2.1%	-2.3%	-3.0%	-3.5%
	8K_seq14	-0.9%	-2.3%	-1.0%	-1.0%	-1.1%
	8K_seq17	-0.6%	-0.2%	-0.1%	-0.6%	-1.6%
	8K_seq18	-0.8%	-0.7%	-0.7%	-0.9%	-1.3%

	DS_store	-0.3%	-0.4%	-0.4%	-0.6%	-0.6%
	2K average	-0.6%	-0.4%	-0.7%	-0.8%	-0.9%
	4K average	-1.5%	-0.9%	-1.3%	-1.8%	-1.9%
	8K average	-1.4%	-1.2%	-1.2%	-1.5%	-1.9%
	All average	**-1.2%**	**-0.8%**	**-1.0%**	**-1.3%**	**-1.6%**

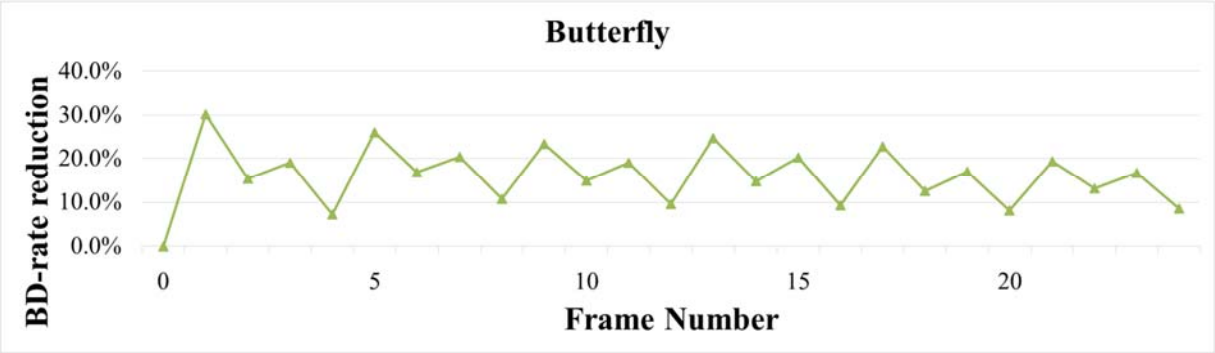
Table 5.5. BD-Bitrate Difference for Pre-Interpolation Test 1 - 5 (Low Delay P, "-" Means Gain).

		Low delay P				
		Test 1	Test 2	Test 3	Test 4	Test 5
2K (1920×1080)	Kimono	-6.7%	-4.1%	-6.3%	-6.8%	-7.1%
	ParkScene	-1.3%	-1.0%	-1.3%	-1.8%	-1.7%
	Cactus	-1.7%	-0.9%	-1.8%	-2.2%	-2.8%
	BasketballDrive	-2.5%	-1.7%	-2.1%	-3.0%	-3.2%
	BQTerrace	-1.2%	-0.5%	-1.4%	-1.9%	-2.8%
4K (3840×2160) Hotel (4096×2048)	chuno_s4	-3.8%	-1.5%	-2.1%	-4.1%	-4.4%
	chuno_s31	-2.1%	-1.6%	-2.2%	-2.1%	-2.1%
	crownrun	-4.7%	-2.1%	-2.0%	-5.4%	-6.5%
	hotel	-1.4%	-0.9%	-1.3%	-2.0%	-2.3%
	Pku_girl	-1.4%	-0.8%	-0.2%	-1.7%	-3.4%
	reed	-3.2%	-1.1%	-1.4%	-3.0%	-3.1%
	4k_seq1	-1.3%	-0.5%	-1.3%	-2.1%	-2.5%
	butterfly	-11.9%	-7.9%	-7.7%	-11.9%	-12.4%
8K (7680×4320)	8K_seq7	-8.3%	-5.3%	-5.4%	-7.8%	-9.3%
	8K_seq12	-11.8%	-7.0%	-6.6%	-11.6%	-12.7%
	8K_seq14	-4.0%	-5.8%	-4.3%	-4.3%	-4.6%
	8K_seq17	-4.0%	-0.4%	0.1%	-3.8%	-7.1%
	8K_seq18	-2.8%	-1.5%	-1.6%	-2.5%	-4.2%
	DS_store	-0.9%	-0.8%	-0.8%	-1.2%	-1.3%
	2K average	-2.6%	-1.6%	-2.6%	-3.1%	-3.5%
	4K average	-3.7%	-2.0%	-2.3%	-4.1%	-4.6%
	8K average	-5.3%	-3.5%	-3.1%	-5.2%	-6.5%
	All average	**-3.9%**	**-2.4%**	**-2.7%**	**-4.1%**	**-4.9%**

We also plot the frame-by-frame results (positive numbers mean gain) for some video sequences in Figure 5.3.



(a) BasketballDrive, LDP coding.



(b) Butterfly, LDP coding.

Figure 5.3. Frame-by-frame BD-rate reduction (positive means gain).

We have the following remarks for the results:

1) The best performance among all tests is the adaptive BLF with three parameter sets (Test 5). The averaging gain is 4.9%, 1.6% and 1.6% for LDP, LDB, and RA, respectively.
2) The mean filter (Test 1), which is a much simpler filter, slightly outperforms the bilateral filter with one parameter set (Test 3), and performs close to the results of the adaptive BLF with two parameter sets (Test 4).

3) From the frame-by-frame results (Figure 5.3), we can see that the bitrate is reduced for all frames, except the first Intra coding frame.

Figure 5.4 shows the filter mode distribution in Test 3 (pre-interpolation bilateral filter with only one parameter) on the “Butterfly” sequence. It can be seen that the central part of the frame choose “no filter” and the remaining part choose bilateral filter. Our observation is that the non-central part of the frame is out of the depth of field (DOF) of the camera. This part is blurred and has relatively heavy noise, while the central part is relatively sharper. Therefore, we can conclude that the bilateral filter can improve the smoothing, blurry, or noisy blocks to find a better match in the motion compensated prediction.

Figure 5.4. Filter mode distribution. (Green blocks: “no filter”, Remaining part: “bilateral filter”)

Table 5.6. Detailed Results of the Post-Interpolation Tests. ("-" Means Gain)

	Test 6			
		Random access	Low delay B	Low delay P
2K (1920×1080)	Kimono	-0.2%	-0.5%	-5.1%
	ParkScene	-0.1%	-0.3%	-1.2%
	Cactus	0.0%	-0.3%	-1.7%
	BasketballDrive	0.0%	-0.5%	-1.7%
	BQTerrace	-0.2%	-0.4%	-2.7%
4K (3840×2160) Hotel (4096×2048)	chuno_s4	-0.7%	-0.8%	-2.3%
	chuno_s31	-0.7%	-0.7%	-2.3%
	crownrun	-0.9%	-0.9%	-3.6%
	hotel	-0.4%	-0.6%	-1.4%
	Pku_girl	-0.1%	-0.3%	-1.5%
	reed	-0.2%	-0.3%	-1.2%
	4k_seq1	-0.4%	-1.0%	-2.8%
	butterfly	-1.4%	-0.6%	-6.1%
8K (7680×4320)	8K_seq7	-0.2%	0.0%	-3.0%
	8K_seq12	-0.4%	0.0%	-5.7%
	8K_seq14	-0.4%	-0.6%	-2.9%
	8K_seq17	0.1%	-0.1%	-0.3%
	8K_seq18	-0.1%	-0.1%	-0.9%
	DS_store	-0.1%	-0.2%	-0.7%
	2K average	-0.1%	-0.4%	-2.4%
	4K average	-0.6%	-0.6%	-2.6%
	8K average	-0.2%	-0.2%	-2.2%
	All average	**-0.3%**	**-0.4%**	**-2.4%**

5.5 CONCLUSIONS

This chapter proposes an RDO-based adaptive bilateral filter scheme in the motion compensated prediction. The simulation results show that we can get significant coding gain (4.9%, 1.6% and 1.6% for LDP, LDB and RA, respectively) with the proposed method.

Chapter 6. CONCLUSIONS AND FUTURE WORK

6.1 CONCLUSIONS

In this dissertation, four methods in video communication and compression are proposed, including:

1) An low-delay error-resilience video coding scheme,
2) A boundary-matching-based intra prediction scheme,
3) An improved intra prediction scheme, and
4) An improved inter prediction scheme.

These methods are implemented into the reference software and the experimental results confirm the effectiveness of these methods.

Low-delay and error-resilient video coding is critical for real-time video chat applications. We present an efficient intra-refresh cycle-size selection model depending on the network packet loss rate and the motion in the video content. We also present a design of the intra-refresh partition and order that are based on the dislocation error, constrained intra-prediction, constrained inter-prediction, and the cycle size. To select the best cycle-size adaptively according to the packet loss rate, we propose a linear model, where no heuristic sequence-specific parameter is needed. Experimental results confirm the effectiveness of the proposed algorithm.

The Intra Block Copy (IntraBC) mode is a newly adopted extension tool in H.265/HEVC for screen content coding (SCC). This scheme works efficiently on screen content coding since usually there are many repeated patterns in the screen content video, and IntraBC can remove the redundancy much better than the traditional intra prediction (where the predictors only come from neighboring pixels). However, it has been shown that the gain of the current IntraBC scheme on

natural content video coding is much smaller than that on the screen content video. We present the fractional IntraBC and the template matching block vector for the current Intra Block Copy scheme in H.265/HEVC Screen Content Extension to make it more suitable for the natural content coding. The gain with IntraBC and our proposed two tools can be up to 5.4%.

Intra coding is essential in both still image and video coding. In the block-based coding scheme, our work provides two main contributions: 1) it discusses the intra prediction weights in the situation when the reference pixels deviate from the model used, and 2) proposes a novel intra prediction scheme using iterative filtering, which is suitable for the distorted reference pixels. The proposed method is used as an extra intra prediction mode in addition to the copying-based H.265/HEVC intra prediction scheme. The theoretical coding gain is used to compare the proposed scheme and the current H.265/HEVC intra prediction scheme. It shows that better gain is achieved by the iterative filtering method when a proper iteration number is selected. The experimental results in the H.265/HEVC reference software also confirm the effectiveness. The BD-bitrate reduction is up to 3.8%.

An RDO-based adaptive bilateral filter scheme in the motion compensated prediction is proposed in this dissertation. Two schemes (pre-interpolation and post-interpolation) can be used, with more than one set of bilateral filter parameters. The simulation results show that we can get significant coding gain (4.9%, 1.6% and 1.6% for LDP, LDB and RA, respectively) with the proposed method.

6.2 POSSIBLE FUTURE WORK

This dissertation provides contributions to delivering low-delay and error-resilience video and improving the video coding efficiency on both intra and inter schemes. Some problems are beyond the scope of this dissertation, which could be studied in the future research on video coding.

6.2.1 *Content-based video coding and mode prediction*

Currently, H.265/HEVC supports many coding modes. New research also continues introducing new coding modes. These coding modes need flags to inform the decoder which introduces significant overheads. If the overhead could be reduced, the coding performance could be significantly improved. Also, to achieve the best coding performance, the H.265/HEVC reference software uses a rate distortion optimization technique to exhaustively search all the combinations of possible coding modes, including prediction modes (intra, inter and skip), partitions in quad-tree structure and prediction directions. The searching process is very time-consuming and makes real-time applications difficult, especially for ultra-high definition videos.

In recent years, machine learning approaches have been successfully used in many fields, including the speech recognition, machine translation, image classification, and face recognition. One possible future is to investigate the possibility of using machine techniques to predict the coding modes based on the features extracted from the video content. If the classification accuracy is high, it can reduce the overheads to improve the coding efficiency and also reduce the computation since only a few sets of coding modes need to be tested.

6.2.2 *Deep learning aided video quality assessment in rate-distortion optimization*

In current video coding standard, the sum of absolute difference (SAD) or sum of square differences (SSD) are used in the rate-distortion optimization process to get the best tradeoff between video quality and bitrate. However, it has been pointed out that SAD and SSD are not good metrics to reflect human's perceptual system. Other reference-based objective video assessment metrics including Structural Similarity Index (SSIM) [42] and Multi-Scale SSIM (MS-

SSIM) [80] are proposed. But our studies show that on some certain data sources, these metrics do not work well either.

One recent work shows that the deep convolutional neural network (CNN) can achieve the state-of-the-art result on the non-reference-based image quality assessment [81]. Some of our preliminary results also show that the CNN can achieve higher correlation with human's mean opinion scores (MOS) compared with SAD and SSIM on the movie dataset. This gives us a hint that the deep learning can achieve better trade-off between bitrate and the quality-of-experience.

BIBLIOGRAPHY

[1] "ISO/IEC 13818-2 and ITU-T Rec. H-262. Information technology - generic coding of moving pictures and associated audio information: Video," 1996.

[2] "ISO/IEC 14496-2, Coding of Audio-Visual Objects - Part 2: Visual," 2001.

[3] "ISO/IEC 14496-10 and ITU-T Rec. H.264, Advanced Video Coding," 2003.

[4] "ISO/IEC 23008-2 MPEG-H Part 2 and ITU-T H.265, Information technology -- High efficiency coding and media delivery in heterogeneous environments -- Part 2: High efficiency video coding," 2013.

[5] "Chromium open-source browser project, VP9 source code." [Online]. Available: http://www.webmproject.org/vp9/. [Accessed: 04-Sep-2015].

[6] "Cisco Visual Networking Index: Forcast and Methodology 2013 – 2018," 2013. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.html. [Accessed: 04-May-2015].

[7] 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. Image Process.*, vol. 22, no. 2, pp. 790–801, Feb. 2013.

[8] A. Saxena and F. C. Fernandes, "DCT/DST-based transform coding for intra prediction in image/video coding.," *IEEE Trans. Image Process.*, vol. 22, no. 10, pp. 3974–81, Oct. 2013.

[9] J.-W. Kang, M. Gabbouj, and C.-C. J. Kuo, "Sparse/DCT (S/DCT) two-layered representation of prediction residuals for video coding.," *IEEE Trans. Image Process.*, vol. 22, no. 7, pp. 2711–22, Jul. 2013.

[10] W. Dai, H. Xiong, X. Jiang, and C. W. Chen, "Structured Set Intra Prediction With Discriminative Learning in a Max-Margin Markov Network for High Efficiency Video Coding.," *IEEE Trans. circuits Syst. video Technol. a Publ. Circuits Syst. Soc.*, vol. 23, no. 11, pp. 1941–1956, Nov. 2013.

[11] C. Wei, H. Chen, M. Song, M.-T. Sun, and K. Lau, "A capture-to-display delay measurement system for visual communication applications," in *2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, 2013, pp. 1–4.

[12] H. Chen, C. Wei, M. Song, M.-T. Sun, and K. Lau, "Capture-to-display delay measurement for visual communication applications," *APSIPA Trans. Signal Inf. Process.*, vol. 4, p. e19, Dec. 2015.

[13] H. Chen, C. Zhao, M.-T. T. Sun, and A. Drake, "Adaptive intra-refresh for low-delay error-resilient video coding," in *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific*, 2014, pp. 1–4.

[14] H. Chen, C. Zhao, M.-T. Sun, and A. Drake, "Adaptive intra-refresh for low-delay error-resilient video coding," *J. Vis. Commun. Image Represent.*, vol. 31, pp. 294–304, Aug. 2015.

[15] H. Chen, Y. Chen, M. Sun, A. Saxena, and M. Budagavi, "Improvements on Intra Block

Copy in Natural Content Video Coding," in *ISCAS 2015*, 2015, pp. 0–3.

[16] C. Tomasi and R. Manduchi, "Bilateral Filtering for Gray and Color Images," *Int. Conf. Comput. Vis.*, pp. 839–846, 1998.

[17] X. Chen, J.-N. Hwang, J. Ritcey, C.-N. Lee, and F.-M. Yeh, "Quality-Driven Joint Rate and Power Adaptation for Scalable Video Transmissions over MIMO Systems," *IEEE Trans. Circuits Syst. Video Technol.*, vol. PP, no. 99, pp. 1–1, 2015.

[18] X. Chen, J.-N. Hwang, C.-Y. Wang, and C.-N. Lee, "A near optimal QoE-driven power allocation scheme for SVC-based video transmissions over MIMO systems," in *2014 IEEE International Conference on Communications (ICC)*, 2014, pp. 1675–1680.

[19] X. Chen, J. N. Hwang, C. N. Lee, and S. I. Chen, "A Near Optimal QoE-Driven Power Allocation Scheme for Scalable Video Transmissions Over MIMO Systems," *IEEE J. Sel. Top. Signal Process.*, vol. 9, no. 1, pp. 76–88, Feb. 2015.

[20] X. Chen, J. N. Hwang, P. H. Wu, H. J. Su, and C. N. Lee, "Adaptive mode and modulation coding switching scheme in MIMO multicasting system," in *2013 IEEE International Symposium on Circuits and Systems (ISCAS2013)*, 2013, pp. 441–444.

[21] Xiang Chen, J. N. Hwang, Chung-Nan Lee, and C. W. Huang, "An efficient CQI feedback resource allocation scheme for wireless video multicast services," in *2013 IEEE Global Communications Conference (GLOBECOM)*, 2013, pp. 1663–1668.

[22] Shao-Shuai Gao and Guo-Fang Tu, "Robust H.263+ video transmission using partial backward decodable bit stream (PBDBS)," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 2, pp. 182–187, Feb. 2003.

[23] S. Kumar, "Error resiliency measure for RVLC codes," *IEEE Signal Process. Lett.*, vol. 13, no. 2, pp. 84–87, Feb. 2006.

[24] R. H. Morelos-Zaragoza, *The Art of Error Correcting Coding*. John Wiley & Sons, 2006.

[25] Tong Gan and Kai-Kuang Ma, "Weighted unequal error protection for transmitting scalable object-oriented images over packet-erasure networks," *IEEE Trans. Image Process.*, vol. 14, no. 2, pp. 189–199, Feb. 2005.

[26] Y. Wang, S. Wenger, J. Wen, and A. K. Katsaggelos, "Error resilient video coding techniques," *IEEE Signal Process. Mag.*, vol. 17, no. 4, pp. 61–82, Jul. 2000.

[27] M. Fleury, I. A. Ali, and M. Ghanbari, "Video Intra Coding for Compression and Error Resilience : A Review," *Recent Patents Signal Process.*, vol. 4, no. 1, pp. 32–43, 2014.

[28] "VideoLAN–x264, the best H.264/AVC encoder." [Online]. Available: http://www.videolan.org/developers/x264.html. [Accessed: 12-Jan-2012].

[29] Z. He, J. Cai, and C. W. Chen, "Joint source channel rate-distortion analysis for adaptive mode selection and rate control in wireless video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 6, pp. 511–523, 2002.

[30] G. Côté and F. Kossentini, "Optimal intra coding of blocks for robust video communication over the Internet," *Signal Process. Image Commun.*, vol. 15, no. 1, pp. 25–34, 1999.

[31] "H.264/AVC Reference Software jm18.6." [Online]. Available: http://iphome.hhi.de/suehring/tml/download/jm18.6.zip. [Accessed: 02-Jun-2014].

[32] R. Zhang, S. L. Regunathan, and K. Rose, "Video coding with optimal inter/intra-mode switching for packet loss resilience," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 6, pp. 966–976, 2000.

[33] Q. Chen, Z. Chen, X. Gu, and C. Wang, "Attention-based adaptive intra refresh for error-prone video transmission," *IEEE Commun. Mag.*, vol. 45, no. 1, pp. 52–60, 2007.

[34] Y. Dhondt, S. Mys, K. Vermeirsch, and R. Van de Walle, "Constrained inter prediction: Removing dependencies between different data partitions," in *Advanced Concepts for Intelligent Vision Systems*, Springer Berlin Heidelberg, 2007, pp. 720–731.

[35] M. M. Hannuksela, Y.-K. Wang, and M. Gabbouj, "Isolated Regions in Video Coding," *IEEE Trans. Multimed.*, vol. 6, no. 2, pp. 259–267, Apr. 2004.

[36] R. M. Schreier and A. Rothermel, "Motion Adaptive Intra Refresh for the H.264 Video Coding Standard," *IEEE Trans. Consum. Electron.*, vol. 52, no. 1, pp. 249–253, 2006.

[37] R. Schreier and A. Rothermel, "Motion adaptive intra refresh for low-delay video coding," in *Proceedings of the International Conference on Consumer Electronics (ICCE 2006)*, 2006.

[38] P. Nunes, D. L. Soares, and F. Pereira, "Automatric and adaptive network-aware macroblock intra refresh for error-reslient h.264/avc video coding," in *ICIP 2009*, 2009, pp. 3073–3076.

[39] M. F. Ali and G. M., "Content-aware intra-refresh for video streaming over lossy links," in *International Conference on Consumer Electronics (ICCE 2012)*, 2012.

[40] A. S. Morion, M. Feury, and G. M., "Prioritized packetization for video with intra-refresh macroblock line," *ICME 2011*, 2011.

[41] G. Bjontegaard, "Calculation of average psnr differences between rd curves," in *VCEG-M33 ITU-T Q6/16*, 2001.

[42] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[43] M. Budagavi and D.-K. Kwon, "AHG8: Video coding using Intra motion compensation," in *Joint Collaborative Team on Video Coding (JCT-VC), JCTVC-M0350*, 2013.

[44] R. Joshi, J. Xu, R. Cohen, S. Liu, Z. Ma, and Y. Ye, "Screen content coding test model 2 (SCM 2)," in *Joint Collaborative Team on Video Coding (JCT-VC), JCTVC-R1014*, 2014.

[45] S. Lee, C. Park, E. Alshina, C. Kim, and K. McCann, "AHG5: Recommended profiling of range extension coding tools," in *Joint Collaborative Team on Video Coding (JCT-VC), JCTVC-Q0051*, 2014.

[46] H. Malvar, "Adaptive run-length/Golomb-Rice encoding of quantized generalized Gaussian sources with unknown statistics," in *DCC 2006*, 2006.

[47] T. K. Tan, C. S. Boon, and Y. Suzuki, “Intra prediction by averaged template matching predictors,” in *4th Annual IEEE Consumer Communications and Networking Conference, CCNC 2007*, 2007.

[48] C. Lan, J. Xu, F. Wu, and G. Shi, “Intra frame coding with template matching prediction and adaptive transform,” in *ICIP 2010*, 2010, pp. 1221–1224.

[49] B. Girod, “Motion-compensating prediction with fractional-pel accuracy,” *IEEE Trans. Commun.*, vol. 41, no. 4, pp. 604–612, 1993.

[50] K. McCann, W.-J. Han, and I.-K. Kim, “Samsung’s Response to the Call for Proposals on Video Compression Technology,” in *oint Collaborative Team on Video Coding (JCT-VC), JCTVC-A124*, 2010.

[51] “Hevc test model (hm-14.0).” [Online]. Available: http://hevc.kw.bbc.co.uk/git/w/jctvc-hm.git/commit/1204fc83b1f917237624e9ec8c3c4d7bebab58d5. [Accessed: 11-Apr-2014].

[52] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand, “Overview of the high efficiency video coding (HEVC) standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, pp. 1649–1668, 2012.

[53] Y. Chen, J. Han, and K. Rose, “A Recursive Extrapolation Approach to Intra Prediction in Video Coding,” in *ICASSP 2013*, 2013, pp. 1734–1738.

[54] S. Li, Y. Chen, J. Han, T. Nanjundaswamy, and K. Rose, “Rate-Distortion Optimization and Adaptation of Intra Prediction Filter Parameters,” in *ICIP 2014*, 2014, pp. 3146–3150.

[55] F. Kamisli, “Recursive Prediction for Joint Spatial and Temporal Prediction in Video Coding,” *IEEE Signal Process. Lett.*, vol. 21, no. 6, pp. 732–736, 2014.

[56] N. Wiener, *The Extrapolation, Interpolation and Smoothing of Stationary Time Series*. 1949.

[57] J. Xu, F. Wu, and W. Zhang, “On the Coding Gain of Intra-Predictive Transforms,” in *ICIP 2009*, 2009, pp. 13–16.

[58] L. Zhang, X. Zhao, S. Ma, Q. Wang, and W. Gao, “Novel intra prediction via position-dependent filtering,” *J. Vis. Commun. Image Represent.*, vol. 22, no. 8, pp. 687–696, 2011.

[59] X. Qi, T. Zhang, F. Ye, A. Men, and B. Yang, “Intra Prediction with Enhanced Inpainting Method and Vector Predictor for HEVC,” in *ICASSP 2012*, 2012, pp. 1217–1220.

[60] Y. Zhang and Y. Lin, “Improving HEVC Intra Prediction with PDE-Based Inpainting,” in *APSIPA 2014*, 2014.

[61] J. Han, A. Saxena, V. Melkote, and K. Rose, “Jointly optimized spatial prediction and block transform for video and image coding,” *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1874–1884, 2012.

[62] T. Zhang, H. Chen, M.-T. Sun, D. Zhao, and W. Gao, “Hybrid angular intra/template matching prediction for HEVC intra coding,” in *2015 Visual Communications and Image Processing (VCIP)*, 2015, pp. 1–4.

[63] C. Lan, X. Peng, J. Xu, and F. Wu, “Intra and inter coding tools for screen contents,” in

Joint Collaborative Team on Video Coding (JCT-VC), JCTVC-E145, 2011.

[64] H. Chen, A. Saxena, and F. Fernandes, "Nearest-neighbor intra prediction for screen content video coding," in *ICIP 2014*, 2014, pp. 3151–3155.

[65] H. Chen, S. Zhu, and B. Zeng, "Design of non-separable transforms for directional 2-D sources," in *2011 18th IEEE International Conference on Image Processing*, 2011, pp. 3697–3700.

[66] H. Chen and B. Zeng, "Design of low-complexity, non-separable 2-D transforms based on butterfly structures," in *2012 IEEE International Symposium on Circuits and Systems*, 2012, pp. 2921–2924.

[67] H. Chen and B. Zeng, "New Transforms Tightly Bounded by DCT and KLT," *IEEE Signal Process. Lett.*, vol. 19, no. 6, pp. 344–347, Jun. 2012.

[68] T. Zhang, H. Chen, M.-T. Sun, A. Saxena, D. Zhao, and W. Gao, "Adaptive transform with HEVC intra coding," in *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, 2015, pp. 388–391.

[69] N. S. Jayant and P. Noll, *Digital coding of waveforms: principles and applications to speech and video*. Englewood Cliffs, NJ: Prentice-Hall, 1984.

[70] T. Natarajan and N. Ahmed, "Performance Evaluation for Transform Coding Using a Nonseparable Covariance Model," *IEEE Trans. Commun.*, vol. 26, no. 2, pp. 310–312, Feb. 1978.

[71] F. Bossen, "Common HM test conditions and software reference configurations (JCTVC-L1100)," in *JCT-VC*, 2013.

[72] I. K. Kim, J. Min, T. Lee, W. J. Han, and J. H. Park, "Block partitioning structure in the HEVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1697–1706, 2012.

[73] T. Nguyen, P. Helle, M. Winken, B. Bross, D. Marpe, H. Schwarz, and T. Wiegand, "Transform Coding Techniques in HEVC," *IEEE J. Sel. Top. Signal Process.*, vol. 7, no. 6, pp. 978–989, 2013.

[74] J. Lainema, F. Bossen, W. J. Han, J. Min, and K. Ugur, "Intra coding of the HEVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1792–1801, 2012.

[75] K. Ugur, A. Alshin, E. Alshina, F. Bossen, W.-J. Han, J.-H. Park, and J. Lainema, "Motion Compensated Prediction and Interpolation Filter Design in H.265/HEVC," *IEEE J. Sel. Top. Signal Process.*, vol. 7, no. 6, pp. 946–956, Dec. 2013.

[76] J. L. Lin, Y. W. Chen, Y. W. Huang, and S. M. Lei, "Motion vector coding in the HEVC Standard," *IEEE J. Sel. Top. Signal Process.*, vol. 7, no. 6, pp. 957–968, 2013.

[77] V. Sze and M. Budagavi, "High throughput CABAC entropy coding in HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1778–1791, 2012.

[78] W. Pu, W.-S. Kim, J. Chen, K. Rapaka, L. Guo, J. Sole, and M. Karczewicz, "Non-RCE1: Inter Color Component Residual Prediction," in *Joint Collaborative Team on Video Coding*

(JCT-VC), JCTVC-N0266, 2013.

[79] B. Girod, “Efficiency analysis of multihypothesis motion-compensated prediction for video coding,” *IEEE Trans. Image Process.*, vol. 9, no. 2, pp. 173–183, 2000.

[80] Z. Wang, E. P. E. P. Simoncelli, and A. C. a. C. Bovik, “Multi-scale structural similarity for image quality assessment,” *IEEE Asilomar Conf. Signals, Syst. Comput.*, vol. 2, no. 1, pp. 1398–1402, 2003.

[81] L. Kang, P. Ye, Y. Li, and D. Doermann, “Convolutional Neural Networks for No-Reference Image Quality Assessment,” in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1733–1740.

APPENDIX A

The 2-D directional Gaussian Markov model in Eq. (4.16) can be rewritten in a linear format,

$$s = \beta_1 t_1 + \beta_2 t_2 ,$$

where

$$s = \log^2 E(x_{i,j} x_{p,q}) ,$$

$$\beta_1 = \log^2 \rho ,$$

$$t_1 = d_1^2(\alpha) ,$$

$$\beta_2 = \eta^2 \bullet \log^2 \rho , \text{ and}$$

$$t_2 = d_2^2(\alpha) .$$

Given the auto-correlation matrix of the mode and the dominating direction angle α, the values s, t_1 and t_2 are known and the β_1 and β_2 can be derived from a linear regression, then the ρ and η are calculated. The training process has following steps:

1) Select the first frame from six sequences: Kimono, Cactus, BasketballDrive, BQTerrace, 4K_Seq1 and Butterfly.
2) For each image, it is divided into non-overlapped 8×8 small patches and the intra mode of each patch is determined via minimizing the SAD between the prediction and the image patch.
3) All patches with the same mode are classified as the "observations" of that mode and the auto-correlation matrix is calculated.
4) Use the least square method to find the β_1 and β_2, and calculate ρ and η.

For some of these modes, when the trained parameter is not a real number, e.g., when β_1 is negative there will be no corresponding real number for ρ, parameters are linearly interpolated

with parameters of its neighboring mode. The final training result is shown in Table 6.1. For simplicity, in analysis within this dissertation, a typical set of parameters $\rho = 0.99$ and $\eta = 5$ (1 for DC and Planar modes) are used.

Table 6.1. Trained Parameters in First-Order Markov 2-D Directional Model

Mode	0	1	2	3	4	5	6
α	0°	0°	45°	39.1°	33.3°	28.0°	22.1°
ρ	0.9927	0.9942	0.9909	0.9945	0.9985	0.9961	0.9930
η	1.0406	1.0167	2.0448	3.4841	16.9953	8.6324	6.0968
Mode	7	8	9	10	11	12	13
α	15.7°	8.8°	3.6°	0°	-3.6°	-8.8°	-15.7°
ρ	0.9904	0.9991	0.9951	0.9911	0.9915	0.9919	0.9924
η	4.4657	38.9562	21.5611	4.1660	4.2770	4.3879	4.4989
Mode	14	15	16	17	18	19	20
α	-22.1°	-28.0°	-33.3°	-39.1°	-45°	129.1°	123.3°
ρ	0.9928	0.9932	0.9936	0.9940	0.9905	0.9934	0.9942
η	4.6099	4.7209	4.8318	4.9428	2.6771	4.1971	5.0094
Mode	21	22	23	24	25	26	27
α	118.0°	112.1°	105.7°	98.9°	93.6°	90°	86.4°
ρ	0.9951	0.9953	0.9959	0.9954	0.9919	0.9944	0.9941
η	5.8217	6.3431	8.5213	7.1970	3.8479	3.5534	6.8005
Mode	28	29	30	31	32	33	34
α	81.1°	74.3°	67.9°	62.0°	56.7°	50.9°	45°
ρ	0.9950	0.9958	0.9967	0.9973	0.9983	0.9928	0.9910
η	6.5934	7.4613	8.9982	11.1011	14.4937	2.8417	2.3281

APPENDIX B

For the distorted reference, the optimal weights are derived by Eq. (26). Given N reference samples, the following approximation holds when σ^2 is large and ρ -> 1.

$$E(\overline{A}\overline{A}^T) \approx \begin{pmatrix} 1+\sigma^2 & 1 & \cdots & 1 \\ 1 & 1+\sigma^2 & \cdots & \vdots \\ \vdots & \vdots & \ddots & 1 \\ 1 & \cdots & 1 & 1+\sigma^2 \end{pmatrix}_{N\times N}$$

$$E(b_i\overline{A}) \approx \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}_{N\times 1}.$$

Hence the optimal weights are approximated as

$$w_{i,\text{opt}} \approx \begin{pmatrix} 1+\sigma^2 & 1 & \cdots & 1 \\ \vdots & 1+\sigma^2 & \cdots & \vdots \\ \vdots & \vdots & \ddots & 1 \\ 1 & \cdots & 1 & 1+\sigma^2 \end{pmatrix}^{-1} \bullet \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$

$$= \frac{1}{\sigma^2+N}\begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}.$$

So when the error is large, the optimal weights of each reference pixels are the same.

VITA

Haoming Chen

Education

- **University of Washington** — Seattle, WA
  Ph.D., Electrical Engineering — *2012 - 2016*
- **Hong Kong University of Science & Technology** — Hong Kong
  M.Phil., Electronic and Computer Engineering — *2010 - 2012*
- **Nanjing University** — Nanjing, China
  B.S., Electrical Engineering — *2006 - 2010*

Research Experience

- **University of Washington** — Seattle, WA
  Research assistant with Prof. Ming-Ting Sun — *9/2012 - 6/2016*
  - Next generation video coding
  - Low-delay error-resilience video coding
  - Capture-to-display delay measurement for Video Chat
- **Apple** — Cupertino, CA
  Intern — *Summer 2015*
  - Image quality assessment using deep convolutional neural network
- **Samsung Research America** — Richardson, TX
  Intern — *Summer 2013*
  - Screen content video coding

Publications

Journal Paper

- **H. Chen**, T. Zhang, M.T. Sun, A. Saxena and M. Budagavi, "Improving Intra Prediction in High Efficiency Video Coding, " *IEEE Transactions on Image Processing*, accepted, Mar. 2016.
- **H. Chen**, C. Wei, M. Song, M.T. Sun and K. Lau, "Capture-to-Display Delay Measurement for Visual Communication Applications, " *APSIPA Transactions on Signal and Information Processing*, Volume 4, 2015.
- **H. Chen**, C. Zhao, M.T. Sun and A. Drake, "Adaptive Intra-Refresh for Low-Delay Error-Resilient Video Coding, " *Journal of Visual Communication and Image Representation*, Vol. 31, pp. 294-304, 2015.

Conference Paper

- T. Zhang, **H. Chen**, M.T. Sun, A. Saxena, D. Zhao and W. Gao, "Adaptive Transform with HEVC Intra Coding, " *APSIPA ASC 2015*, 2015.
- T. Zhang, **H. Chen**, M.T. Sun, D. Zhao and W. Gao, "Hybrid Angular Intra/Template Matching Prediction for HEVC Intra Coding, " *IEEE International Conference on Visual Communications and Image Processing (VCIP-2015)*, 2015.
- **H. Chen**, Y. Chen, M.T. Sun, A. Saxena and M. Budagavi, "Improvements on Intra Block Copy in Natural Content Video Coding, " *IEEE International Symposium on Circuits and Systems (ISCAS-2015)*.
- **H. Chen**, A. Saxena and F. Fernandez, "Nearest-neighboring intra prediction for screen content video coding, " *the 21th IEEE International Conference on Image Processing (ICIP-2014)*. Top 10% paper.
- **H. Chen**, C. Zhao, M.T. Sun and A. Drake, "Adaptive Intra-Refresh for Low-Delay Error-Resilience Video Coding," *APSIPA ASC* 2014, December 2014.
- C. Wei, **H. Chen**, M. Song, M.T. Sun and K. Lau, "A capture-to-display delay measurement system for visual communication applications", *APSIPA ASC* 2013, Kaohsiung, Taiwan, Oct. 29-31, 2013

Patent

- Ankur Saxena, **Haoming Chen** and Felix Carlos Fernandes, "Method for intra prediction improvements for oblique modes in video coding," US Patent 20150016516 A1, Jan 15, 2015.