Graduate School ETD Form 9
(Revised 12/07)

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Meilin Yang

Entitled
Multiple Description Video Coding with Adaptive Error Concealment

For the degree of Doctor of Philosophy

Is approved by the final examining committee:

EDWARD J. DELP, Co-Chair
Chair

MARY L. COMER, Co-Chair

CHARLES A. BOUMAN

YUNG-HSIANG LU

To the best of my knowledge and as understood by the student in the *Research Integrity and Copyright Disclaimer (Graduate School Form 20)*, this thesis/dissertation adheres to the provisions of Purdue University's "Policy on Integrity in Research" and the use of copyrighted material.

Approved by Major Professor(s): MARY L. COMER, Co-Chair

Approved by: M. R. Melloch 4/17/2012
Head of the Graduate Program Date

Graduate School Form 20
(Revised 9/10)

PURDUE UNIVERSITY
GRADUATE SCHOOL

Research Integrity and Copyright Disclaimer

Title of Thesis/Dissertation:
Multiple Description Video Coding with Adaptive Error Concealment

For the degree of Doctor of Philosophy

I certify that in the preparation of this thesis, I have observed the provisions of *Purdue University Executive Memorandum No. C-22,* September 6, 1991, *Policy on Integrity in Research.**

Further, I certify that this work is free of plagiarism and all materials appearing in this thesis/dissertation have been properly quoted and attributed.

I certify that all copyrighted material incorporated into this thesis/dissertation is in compliance with the United States' copyright law and that I have received written permission from the copyright owners for my use of their work, which is beyond the scope of the law. I agree to indemnify and save harmless Purdue University from any and all claims that may be asserted or that may arise from any copyright violation.

Meilin Yang
Printed Name and Signature of Candidate

04-24-2012
Date (month/day/year)

*Located at http://www.purdue.edu/policies/pages/teach_res_outreach/c_22.html

MULTIPLE DESCRIPTION VIDEO CODING WITH ADAPTIVE ERROR CONCEALMENT

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Meilin Yang

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

May 2012

Purdue University

West Lafayette, Indiana

To my parents, Hongbo Yang and Zhuoyan Sun;
To my grandparents, Tianzhu Yang, Shiwen Cheng, Konglu Sun and Xiuzhi Yuan;
To my husband, Mengtian Sun.

ACKNOWLEDGMENTS

I have so many names in my mind when I am writing the acknowledgements. I could never have gone this far without their friendship and help.

First of all, I would like to thank my advisors, Professor May L. Comer and Professor Edward J. Delp for their encouragement, support and confidence in me. They made my dream come true. I appreciate all the academic related and non-academic related conversations with them. It is amazing to be their student and be a member of the Video and Image Processing (VIPER) lab.

I would like to thank my other committee members, Professor Charles Bouman and Professor Yung-Hsiang Lu for their advice, encouragement and insights.

I would like to thank Cisco System for their sponsorship of the research in this dissertation.

I would like to thank all my brilliant colleagues for their support and friendship: Dr. Golnaz Abdollahian, Ziad Ahmad, Dr. Marc Bosch, Dr. Hsiao-Chiang Chuang, Neeraj Gadgil, Andrew W. Haddad, Ye He, Dr. Nitin Khanna, Dae Woo Kim, Dean King-Smith, Kevin Lorenz, Dr. Ying Chen Lou, Aravind Mikkilineni, Dr. Ka Ki Ng, Albert Parra, Dr. Satyam Srivastava, Chang (Joy) Xu, Bin Zhao, Dr. Rong Zhang, Huixi Zhao and Dr. Fengqing (Maggie) Zhu. I would like Professor Yu Charlie Hu and Rohan Gandh from Distributed Systems and Networking lab for the cooperation. Thanks to all the visitors in the lab that I have interacted with: Murat Birinci, Professor Fernando Diaz de Maria, and Professor Moncef Gabbouj.

I would like to thank my parents for their enduring support and encouragement. They have made many sacrifices throughout the years while I have been away from home pursuing my academic career. I thank them for giving me life, teaching me how to be a good person and encouraging me to do what I want to do. Finally, I would

like to thank my husband, Mengtian Sun, for his love, support, understanding and encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

ABBREVIATIONS

AVC	Advanced Video Coding
CABAC	Context-Adaptive Binary Arithmetic Coding
CAVLC	Context-Adaptive Variable-Length Coding
CBP	Coded Block Pattern
CCITT	International Telegraph and Telephone Consultative Committee
CGS	Coarse-granular Scalability
CIF	Common Intermediate Format
CR	Conditional Replacement
DCT	Discrete Cosine Transform
DP	Data Partitioning
FEC	Forward Error Coding
FIR	Finite Impulse Response
FGS	Fine-granular Scalability
FMO	Flexible Macroblock Ordering
GOP	Groups Of Pictures
IDR	Instantaneous Decoding Refresh
ISO	International Organization for Standardization
KLT	Karhunen-Loeve Transform
LOT	Lapped Orthogonal Transform
ITU	International Telecommunication Union
JM	Joint Model
JVT	Joint Video Team
MB	Macroblock
MBAmap	Marcoblock to Slice Map

MC	Motion Compensation
MCTF	Motion-compensated Temporal Filtering
MCP	Motion-Compensated Prediction
MDC	Multiple Description Coding
MDSQ	Multiple Description Scalar Quantizer
MDSVC	Multiple Description Scalable Video Coding
MDVC	Multiple Description Video Coding
ME	Motion Estimation
MPEG	Moving Picture Experts Group
MSE	Mean Square Error
NAL	Network Abstraction Layer
OQ	Objective Quality
PCT	Pair-wise Correlating Transform
PPS	Picture Parameter Set
PR	Progressive Refinement
PSNR	Peak Signal-to-Noise Ratio
QP	Quantization Parameter
RBSP	Raw Byte Sequence Payload
RD	Rate Distortion
RTP	Real Time Protocol
SAMCoW	Scalable Adaptive Motion Compensated Wavelet
SDC	Single Description Coding
SI	Switching I
SNR	Signal-to-noise Ratio
SP	Switching P
SQ	Subjective Quality
SVC	Scalable Video Coding
UDP	User Datagram Protocol
VCEG	Video Coding Experts Group

VCL	Video Coding Layer
VLC	Variable-Length Code

ABSTRACT

Yang, Meilin. Ph.D., Purdue University, May, 2012. Multiple Description Video Coding with Adaptive Error Concealment. Major Professors: Mary L. Comer and Edward J. Delp.

With the development of 3G/4G and WiFi networks, there has been a growing demand for video delivery over wireless channels. This increase of video traffic places a significant challenge on efficient, reliable and adaptable video coding techniques. Multiple description coding (MDC) is one of the most efficient error resilient video coding methods especially for real-time applications when retransmission is unacceptable. Usually, MDC uses two equally important descriptions so that each description can be decoded independently at an acceptable decoding quality. The decoding quality is improved when both descriptions are received. However, in applications with scalable, multicast and P2P environments, it is advantageous to use more than two descriptions.

In this dissertation, we have developed novel MDC methods with adaptive error concealment. We proposed a MDC partition architecture with four descriptions based on temporal and spatial correlations. In this method, the MDC partition is done before the prediction so that each description has an independent prediction loop. To improve the error robustness, we developed several adaptive error concealment methods. One method is frame level based on error tracking. It takes into account the distortion due to error concealment and error propagation. Other methods are on the macroblock level based on foreground-background and distortion mapping respectively. We also investigated a MDC partition architecture with better coding efficiency and error robustness. In this method, spatial partition is done after the prediction so that each of the two descriptions share the same prediction loop. A

Gilbert model is used for the channel for burst packet loss simulation. Experimental results demonstrate the efficacy of our proposed methods.

1. INTRODUCTION

With the development of 3G/4G and WiFi networks, there has been a growing demand for video delivery over wireless channels. This increasing video traffic along with error prone wireless channels and heterogenous clients place significant challenges on efficient, reliable and adaptable video coding techniques.

A typical video transmission system consists of a source encoder, a channel encoder, the transmission channel, a channel decoder and source decoder as shown in Figure 1.1.

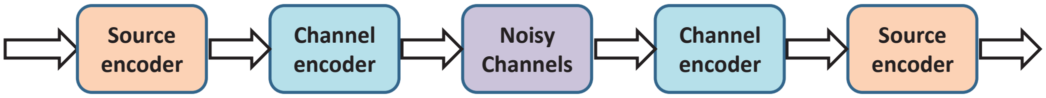


Fig. 1.1. A Typical Video Communication System.

Video source coding aims at representing a video sequence by the lowest data rate (bits/second) for a given image quality [1]. Video compression is achieved by reducing or eliminating spatial redundancy and temporal redundancy in the video sequence [2,3]. Statistical correlations within a video frame is used to reduce the spatial redundancy and statistical correlations among the neighboring video frames can be used to reduce the temporal redundancy. In addition, orthogonal transformations such as the Discrete Cosine Transform (DCT) can concentrate the signal energy in spatial frequency regions and therefore further reduce the redundancy [4]. To ensure the inter-operability between different manufacturers and devices, a series of video coding standards have been developed with the growing requirements of video applications [3,5].

A defining characteristic of a wireless channel is the variations of the “channel strength” over time and frequency [6]. This can cause packet loss during video transmission. As previously mentioned, video compression is achieved by exploiting both temporal and spatial correlations. This strategy induces a large amount of dependency among transmitted data packets. As a result, a lost packet corresponding to compressed data in a frame can cause the reconstructed frame at the decoder to be different from the one at the encoder. In a typical motion compensated video encoder, the encoder uses an error free frame as a reference when encoding future frames while the reference frame at the decoder has errors, these errors can propagate to future decoded frames until a new reference frame is used (instantaneous decoding refresh (IDR) fame). This mismatch and error propagation phenomenon is known as “drift” and can result in severe loss of decoded video quality [7, 8]. Therefore error resilient video coding, i.e. introducing error resilience at the encoder, is very important.

In this chapter, we first review the start-of-art video coding standard, H.264/AVC, and its transmission over networks. We then review error resiliency schemes for H.264 and recent advanced error resilient video coding methods and their potential applications.

1.1 Overview of H.264/AVC Standard

A video coding standard allows inter-operability between different manufacturers and devices [9]. Video coding standards only specifies the syntax and semantics of a encoded bitstream, i.e. how a decoder works. They do not define the encoder. Since the 1990s, a series of video coding standards have been developed with the growing requirements of video applications. Two organizations play the most important roles in the development of these standards: the Video Coding Experts Group (VCEG) and the Moving Picture Experts Group (MPEG). VCEG is part of the Telecommunication Standardization Sector of International Telecommunications Union (ITU-T), and the corresponding standards are named as “H.26x.” MPEG is part of the International

Organization for Standardization (ISO/ IEC), and the corresponding standards are labeled by “MPEG-x”. The timeline of video coding standards is shown in Figure 1.2.

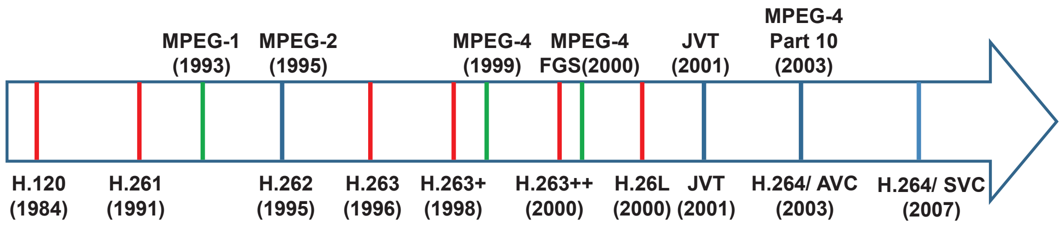


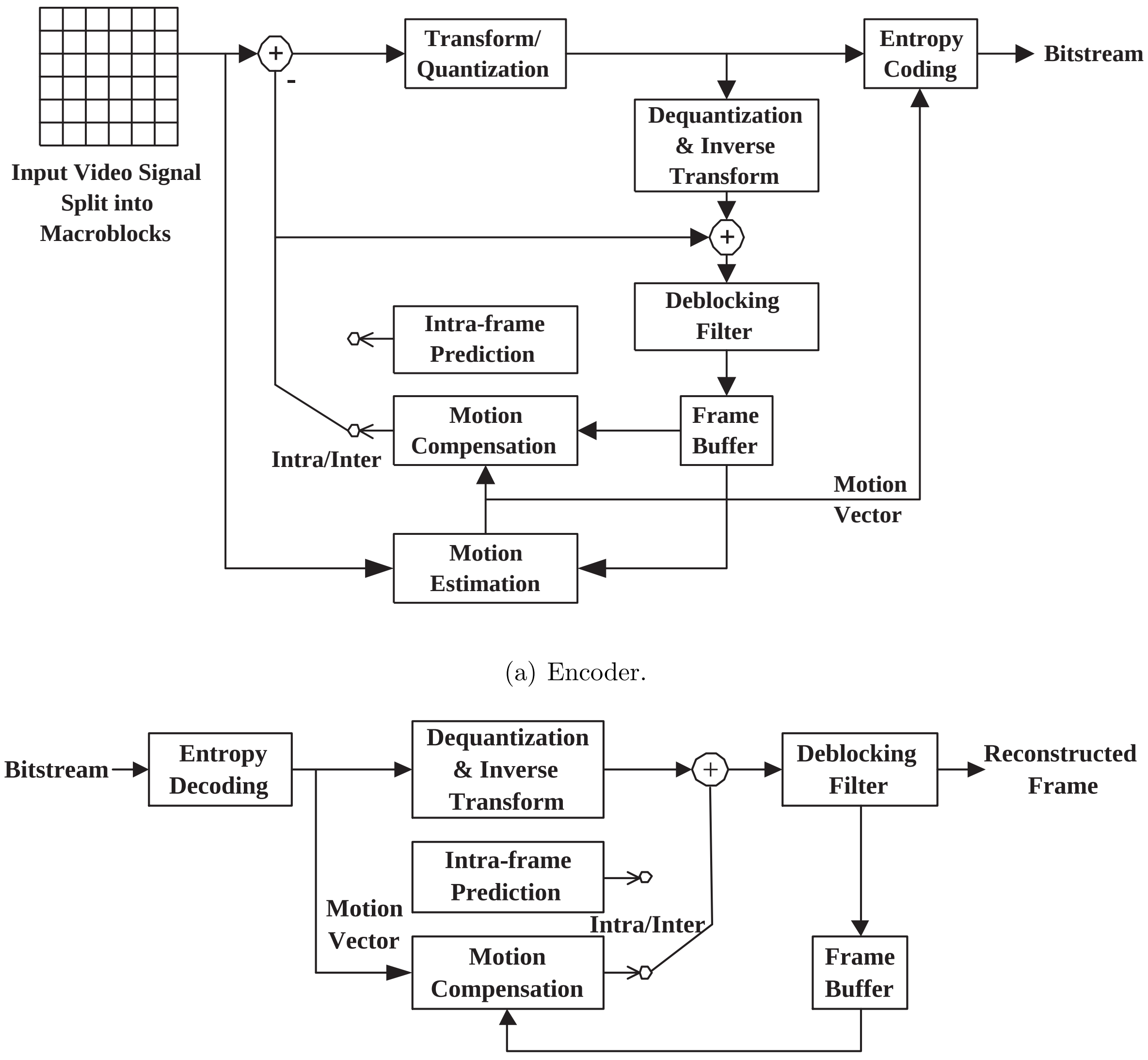
Fig. 1.2. Timeline of Video Coding Standards (MPEG-2/H.262 and MPEG-4 Part 10 AVC/H.264 are joint projects).

In video coding standard development, each new standard is built on its preceding generation. Since 2000, ITU-T and ISO have been working together on video coding standards and formed a group: Joint Video Team (JVT). In 2004, they jointly approved H.264 and MPEG-4 Part 10: Advanced Video Coding (referred to as H.264/AVC) [7, 10–12]. The reference software for H.264/AVC is known as the Joint Model (JM) and is available on online [13]. H.264/AVC has achieved a significant improvement over all the prior standards and is used in wireless, IP networks and digital cinema.

As all the previous standards, H.264/AVC adopts a block-based hybrid video coding structure as shown in Figure 1.3 [7]. An input video sequence is processed in units of 16×16 blocks known as a macroblock (MB). A macroblock contains 16×16 luma samples and the associated 8×8 chroma samples. Each macroblock is encoded in intra or inter mode. Intra prediction uses the previously coded spatially neighboring blocks as references and inter prediction uses the previously coded temporally neighboring blocks as references. For inter prediction, the motion vector is generated through motion estimation which represents the displacement between the current block and its reference block. After motion compensation, the prediction error block is transformed and quantized. In the forward path, the quantized transform coefficients are reordered and entropy encoded and passed to a Network Abstraction Layer (NAL) along with side information for transmission or storage. At the same time, in the reconstruction path, the quantized transform coefficients are inverse quantized and inverse transformed. The reconstructed block is passed to deblocking filter as a reference for the following blocks. At the decoder side, a compressed bitstream from the NAL is entropy decoded, inverse quantized, inverse transformed and the reconstructed block is passed to deblocking filter as a reference for decoding its following blocks.

In H.264, a video frame is coded as one or more slices. Each slice contains a number of macroblocks up to the total number of macroblocks in a frame. Figure 1.4 shows an example of a frame that contains three slices. Each slice is self-contained and can be decoded without the use of information from other slices [7]. This strategy can efficiently control error propagation. There are three common types of slices. I slices may contain only I macroblocks. P slices may contain P and I macroblocks. And B slices may contain B and I macroblocks.

In H.264/AVC, there are 9 intra prediction modes for 4×4 luma blocks and 4 prediction modes for 16×16 luma blocks. All the prediction modes are implemented and the encoder selects the one with the smallest difference between the prediction block and original block. Figure 1.5 shows the prediction modes for 4×4 blocks [14]. Mode 0 is vertical prediction mode. The upper samples A, B, C, D are extrapolated vertically. Mode 1 is horizontal prediction mode. The left samples I, J, K, L are extrapolated horizontally. Mode 2 is DC prediction mode. All samples in prediction block are predicted by the mean of samples A, B, C, D and I, J, K, L. Mode 3 is diagonal down-left prediction mode. The samples are interpolated at 45 degree between lower-left and upper-right. Mode 4 is diagonal down-right prediction mode. The samples are extrapolated 45 degree down and to the right. Mode 5 is vertical-right prediction mode. Extrapolation is at approximately 26.6 degree to the left of vertical. Mode 6 is horizontal-down prediction mode. Extrapolation is at an angle of about 26.6 below horizontal. Mode 7 is vertical-left prediction mode. The extrapolation



(a) Encoder.

(b) Decoder.

Fig. 1.3. Basic Coding Structure in H.264.

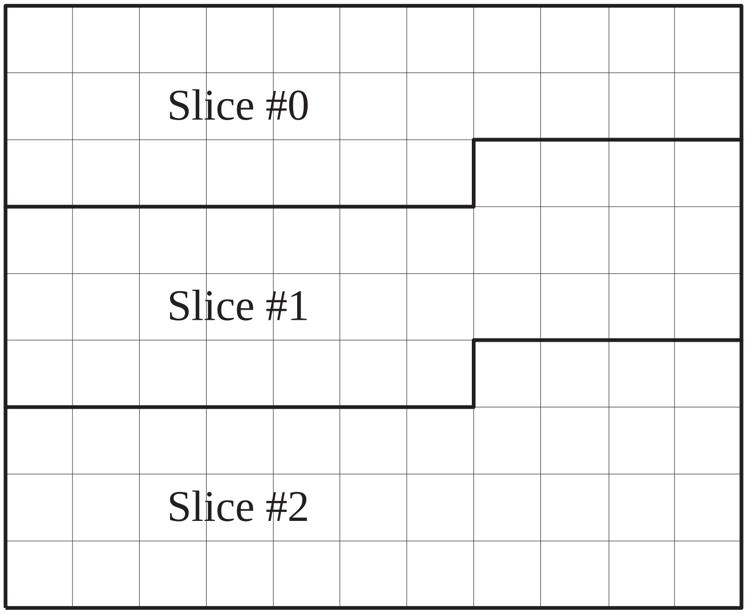
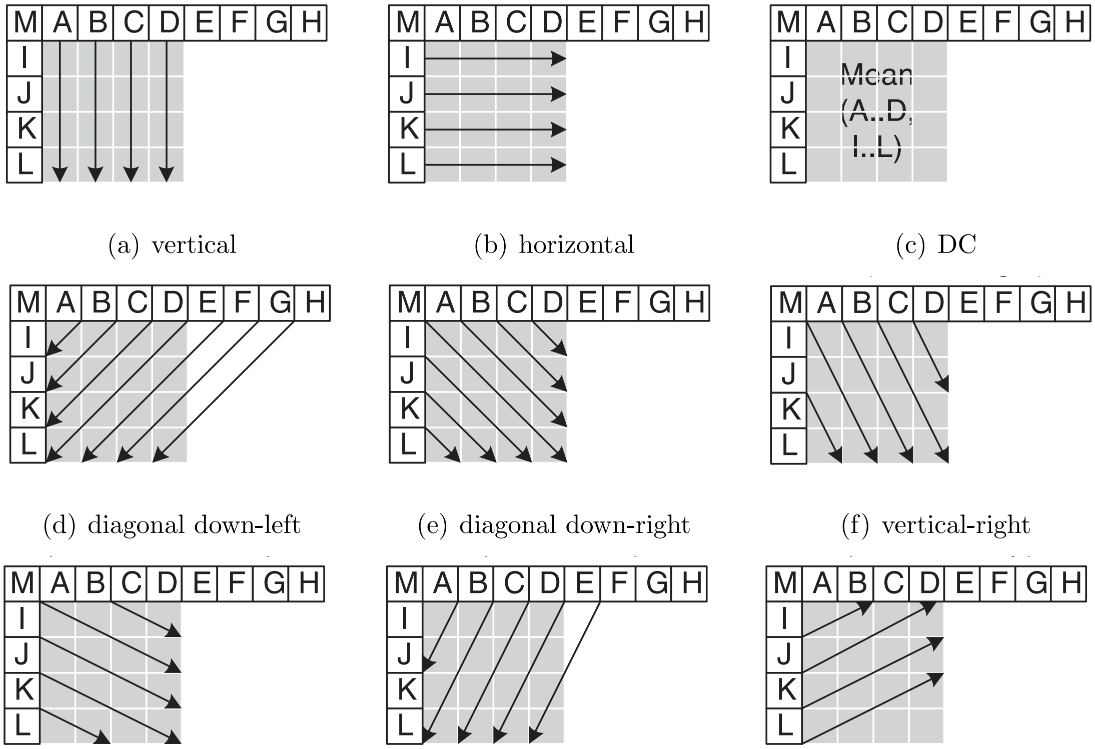


Fig. 1.4. An Example Of A Video Frame That Contains Three Slices.

is at an angle of approximately 26.6 degree to the right of the vertical. Mode 8 is horizontal-up prediction mode. The interpolation is at an angle of approximately 26.6 degree above horizontal. Figure 1.6 shows the prediction modes for 16 × 16 blocks [14]. Mode 0 is vertical prediction mode. The extrapolation is obtained from the upper samples (H). Mode 1 is horizontal prediction mode. The extrapolation is obtained from the left samples (V). Mode 2 is DC prediction mode. The prediction block is formed by the mean of the upper and left-hand samples (H + V). Mode 3 is Plane prediction mode. A linear plane function is fitted to the upper and left-hand samples (H and V).



(a) vertical

(b) horizontal

(c) DC

(d) diagonal down-left

(e) diagonal down-right

(f) vertical-right

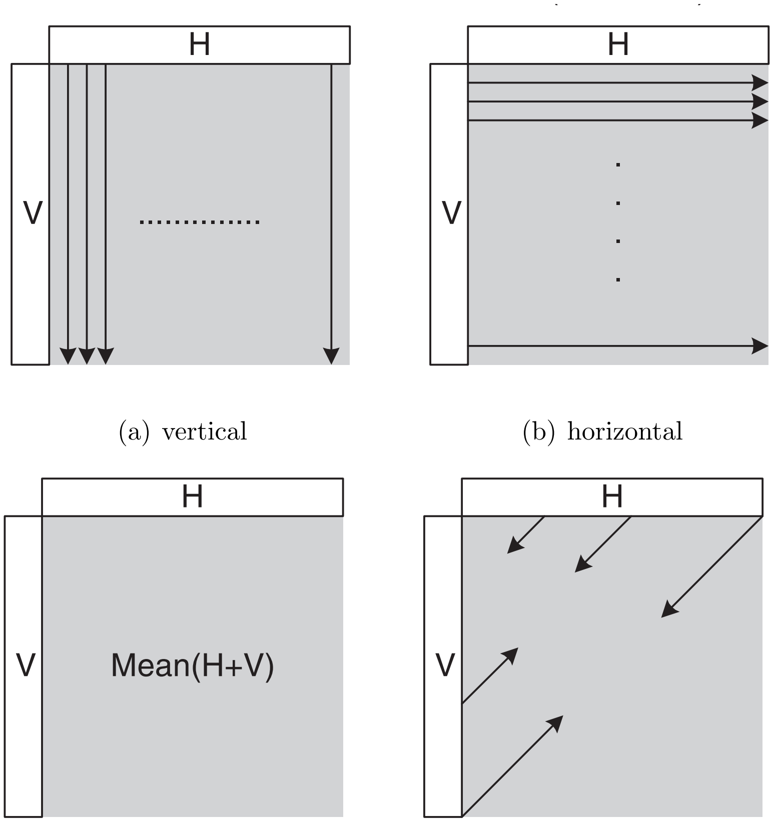
(g) horizontal-down

(h) vertical-left

(i) horizontal-up

Fig. 1.5. Intra 4 × 4 Prediction Modes in H.264.

There are 7 macroblock partition sizes as shown in Figure 1.7. The size of motion estimation is tree structured. A 16 × 16 block can be split as one 16 × 16 block, two 16 × 8 blocks, two 8 × 16 blocks, or four 8 × 8 blocks. If 8 × 8 macroblock partition is



(a) vertical

(b) horizontal

(c) DC

(d) plane

Fig. 1.6. Intra 16 × 16 Prediction Modes in H.264.

chosen, it can be further partitioned as one 8×8 block, two 8×4 blocks, two 4×8 blocks, or four 4×4 blocks. Each block has a motion vector. All the partition sizes are implemented and the encoder selects the one with the smallest rate distortion (RD) cost. The RD cost function is $D+\lambda R$, where D is the distortion, R is the bits used for coding, and λ is a Lagrange parameter. The motion vectors can have quarter-pixel resolution [15]. If both the horizontal and vertical motion vector are integers, the prediction values are the corresponding pixel values of the reference block. If both of the motion vectors are not integers, the prediction values are obtained by interpolation between adjacent samples in the reference block. For half-pixel positions, a 6-tap Finite Impulse Response (FIR) filter is used for the interpolation. For example, in Figure 1.8, b at a half-pixel position is obtained by:

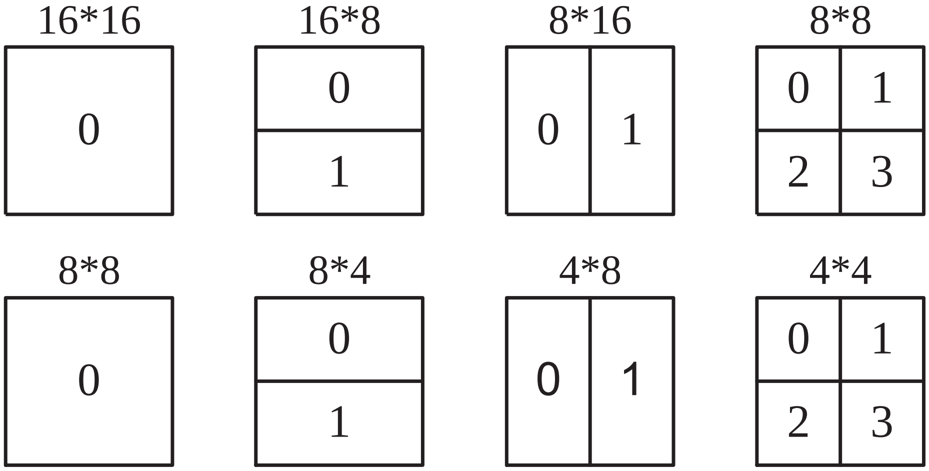


Fig. 1.7. Segmentation of the Macroblock for Motion Compensation.

$$b = round((E - 5F + 20G + 20H - 5I + J)/32) \tag{1.1}$$

Where *round* is a the rounding function.

The prediction values at quarter-pixel positions are obtained as the average of the pixel values at the integer positions and half-pixel positions. For example, in Figure 1.8, a at a quarter-sample position is obtained by:

$$a = round((G + b)/2) \tag{1.2}$$

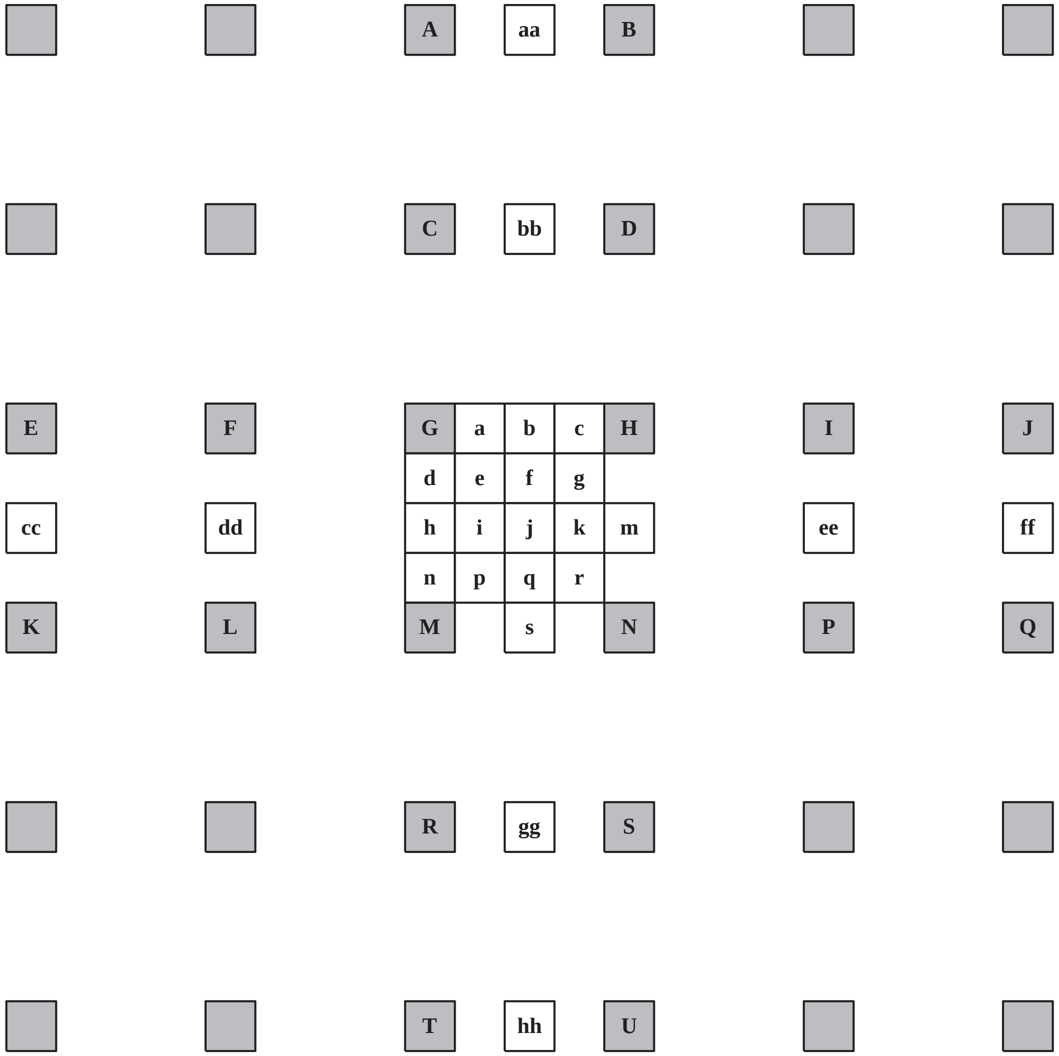


Fig. 1.8. Filtering for Fractional-Pixel Accurate Motion Compensation.

There are three transforms [16] used on H.264/AVC. A Hadamard Transform is used for 4 × 4 luma DC coefficients in intra prediction mode. A DCT-based integer transform is used for all other 4 × 4 luma coefficients. A Hadamard transform is used for 2 × 2 chroma DC coefficients. The integer transform is an approximation of the DCT and has many advantages. Only additions, subtractions and bit shifts are needed for implementation with zero mismatch between the encoder and decoder inverse transform. In addition, the scaling multiplication of the transform can be intergrated into the quantizer which saves the number of multiplications needed. The procedures of transform, quantization, inverse quantization and inverse transform are as follows:

1) Forward transform:

$$W = C_f X C_f^T = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} [X] \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix} \tag{1.3}$$

Where X is the 4×4 residue data.

2) Forward transform post-scaling and quantization:

$$Z = W \times round(PF/Qstep) \tag{1.4}$$

When $Qstep$ is the quantizer and PF is the scaling factor that depends on the positions as shown in Table 1.1.

Table 1.1
Scaling Factors

Position	PF
$(0,0)$,$(2,0)$,$(0,2)$,$(2,2)$	$a^2 = \frac{1}{4}$
$(1,1)$,$(1,3)$,$(3,1)$,$(3,3)$	$\frac{b^2}{4} = \frac{1}{10}$
Other positions	$\frac{ab}{2} = \frac{1}{4}\sqrt{\frac{2}{5}}$

3) Inverse quantization and inverse transform pre-scaling:

$$W' = Z \times Qstep \times PF \times 64 \tag{1.5}$$

4) Inverse core transform:

$$X' = C_i^T W' C_i \tag{1.6}$$

Where $C_i = \begin{bmatrix} 1 & 1 & 1 & \frac{1}{2} \\ 1 & \frac{1}{2} & -1 & -1 \\ 1 & -\frac{1}{2} & -1 & 1 \\ 1 & -1 & 1 & -\frac{1}{2} \end{bmatrix}$

5) Inverse transform post-scaling:

$$X'' = round(X'/64) \tag{1.7}$$

Where X'' is the reconstructed version of X after transform, quantization, inverse quantization and inverse transform.

There are two entropy coding methods possible in H.264/AVC: context-adaptive variable-length coding (CAVLC) and context-adaptive binary arithmetic coding (CABAC) [17]. Both methods take advantage of context-based adaptivity. CAVLC assigns shorter codes for those which are more frequently used. Integer number of bits are used to express each symbol. For CABAC, context model is built on the statistics of already coded syntax elements. Besides, CABAC supports non-integer number of bits per symbol. Therefore, CABAC typically uses $5\% - 15\%$ less bits than CAVLC.

In H.264/AVC, a deblocking filter is used to each decoded macroblock to reduce block distortion [18]. This strategy has two advantages. First, it smooths block edges and thus improves the visual quality. Second, it improves the coding efficiency since the deblocked macroblocks have better quality than the blocky ones.

H.264/AVC supports multiple reference frames for motion-compensated prediction. An example is shown in Figure 1.9. The number of reference frames is specified in the header. The weighted prediction using multiple reference frames provides better performance than a single reference frame [19].

In summary, these advanced techniques improve previous tools resulting in H.264/ AVC having approximately 50% of the data rate with similar perceptual quality in comparison to H.263 and MPEG-4 [7].

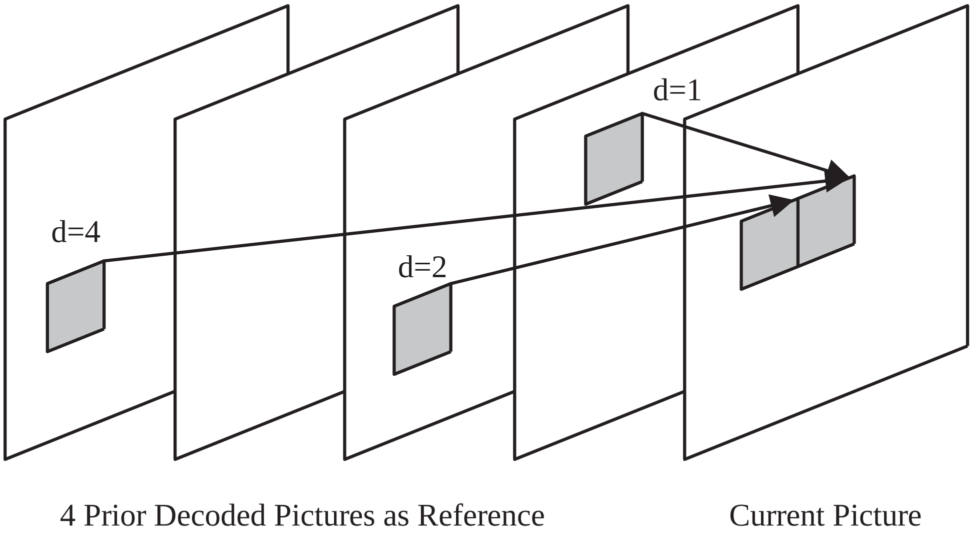


Fig. 1.9. Multiframe Motion Compensation.

1.2 Transmission of H.264/AVC Data Stream

In H.264 a distinction is made between the Video Coding Layer (VCL) and the Network Abstraction Layer (NAL) [20]. The VCL contains the representation of the coded video data and is mapped to NAL units before storage or transmission. The NAL defines the interface between the video coding layer and the transport layer. A coded video sequence consists of NAL units that can be stored in a file or can be transmitted over a packet-based network or circuit switched channel. Each NAL unit contains a Raw Byte Sequence Payload (RBSP) as shown in Figure 1.10. The header of the NAL unit denotes the type of RBSP unit and the RBSP data represents the coded video data. The types of the RBSP include:

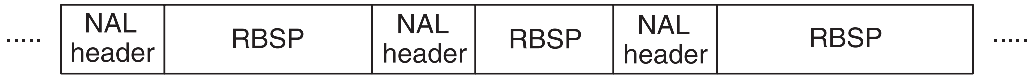


Fig. 1.10. Sequence of NAL Units.

1) Parameter Set: global parameters for a sequence such as frame dimensions, video format, macroblock allocation map.

2) Supplemental enhancement information: side information that is not essential for decoding the video sequence.

3) Picture delimiter: boundary between video frames.

4) Coded slice: header and data for a slice.

5) Data partition A, B or C: it is useful for error resilient decoding. Partition A contains header data for all macroblocks in the slice. Partition B contains intra coded data. Partition C contains inter coded data.

6) End of Sequence: indicates that the next frame is an IDR frame.

7) End of Stream: indicates that there are no more frames in the data stream.

8) Filter data: contains dummy data that is not essential for decoding the video sequence.

For a packet-based network, each NAL unit is carried in a packet and is reordered into the correct sequence before decoding. For a circuit-switched channel, a delimiter is placed before each NAL unit to ensure the decoder can find the start of a NAL unit. The transport and encapsulation of NAL units for different transport systems is shown in Figure 1.11 [21]. These transport mechanisms such as Real Time Protocol and User Datagram Protocol (RTP/UDP) [22], H.320 [23] and MPEG-2 Systems [24] are outside the scope of the H.264/AVC standardization.

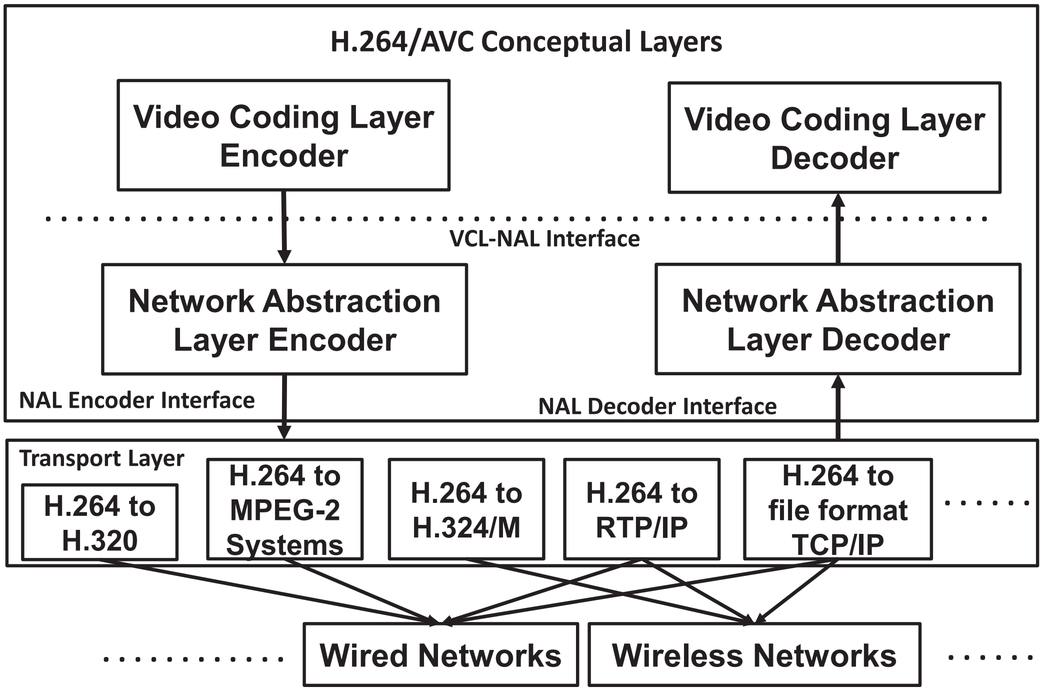


Fig. 1.11. H.264/AVC Standard in Transport Environment.

1.3 Overview of Error Resilient Video Coding

As has been discussed, video compression is achieved by reducing spatial and temporal redundancy. This strategy inevitably induces a large amount of dependency among the transmitted packets. As a result, a lost packet corresponding to a number of macroblocks in a frame can cause the reconstructed frame at the decoder to be different from the one at the encoder. The encoder uses the error free frame as a reference when encoding future frames however the reference frame at the decoder has errors. These errors can propagate to the future decoded frames until the instantaneous decoding refresh (IDR) fame. In addition, variable length coding (VLC) makes the bitstream vulnerable to errors. A single byte error can result in desynchronization making the remaining bitstream undecodable until the next synchronization position [25]. Therefore, error resilient and error concealment methods are indispensable especially for video delivery over unreliable channels such as wireless networks [26, 27]. In this section, we first review error resiliency schemes in H.264/AVC. We then describe recent advanced error resilient methods including scalable video coding (SVC) and multiple description video coding (MDC).

1.3.1 Error Resiliency Techniques in H264/AVC Standard

The error resiliency schemes in H.264/AVC are implemented in the VCL [28]. In this section, we introduce three schemes: data partitioning (DP), flexible macroblock ordering (FMO) and switching P (SP) and switching I (SI) slices.

Data Partitioning

Since some syntax elements of the bitstream such as motion vectors are more important than others, H.264/AVC allows partitioning a slice into up to three different partitions for unequal error protection. Each partition is encapsulated into a separate NAL packet. This scheme is known as data partition (DP). DP A contains the header information such as MB types, quantization parameters and motion vectors, which are the most important part of data in a slice. DP B contains intra coded block patterns (CBPs) and transform coefficients of I blocks, which are the second important part of data in a slice. DP C contains inter CBPs and transform coefficients of P blocks which are the least important part. Both DP B and DP C depend on DP A, but are independent of each other. DP A is the smallest partition of a coded slice which makes it feasible to be transmitted with the highest protection. DP C contains a very large amount of data which should be transmitted with the lowest or no protection. Advanced DP schemes to further improve the performance can be found in [29, 30].

Flexible Macroblock Ordering

A slice group is a subset of a coded picture containing a certain number of macroblocks. MBs are coded in raster scan order within a slice group. Therefore, if a picture is coded in one slice, then all the MBs are coded in raster scan order which is prone to error accumulation. Flexible macroblock ordering(FMO) is an error resilient tool defined in H.264/AVC that allows grouping the macroblocks in a non-raster scan order. In this way, possible errors are scattered throughout the entire frame and error accumulation is constrained to a limited region. The allocation of macroblocks is determined by a marcoblock to slice mapping (MBAmap). H.264/AVC supports seven types of FMO maps. Six are predefined maps, and the other one is the most general type, known as the explicit slice group map, with the entire MBAmap is defined in a picture parameter set (PPS).

The six predefined MBAmaps are illustrated in Figure 1.12 [31]. In type 0, each slice group has a maximum number of macroblocks that are coded in raster scan order before another slice group is started. Type 1 is also known as scattered slices or dispersed slices. In type 2, a picture is consists of one or more rectangular slice groups and a background. The rectangular slice groups can have overlaps but each overlapped area only belongs to one slice group. Type 3 to 5 are dynamic types that only contain two slice groups with one slice growing and the other shrinking. In type 3, one slice group evolves from the center in clockwise or counterclockwise and the other slice group contains the rest of the macroblocks within the picture. In type 4, the growing slice group evolves from the top in raster scan order. Type 5 is similar to type 4, in which the growing slice group evolves from the left in raster scan order.

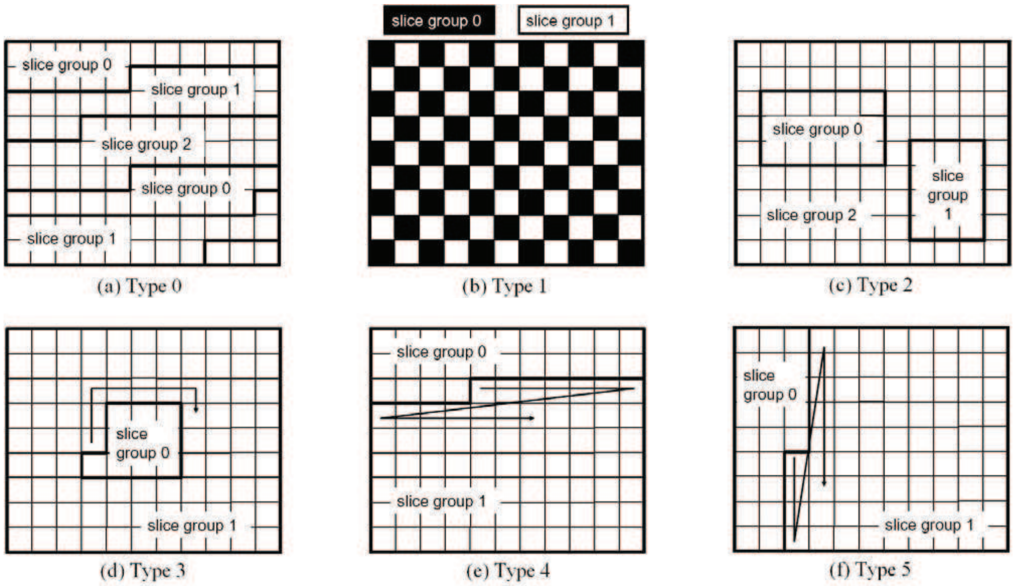


Fig. 1.12. Predefined MBAmaps [31].

The seventh type of MBAmaps is a general type that can be designed by the following steps:

1) Parameter specification: find a parameter to quantify the importance of a macroblock.

2) Macroblock classification: classify the macroblocks to slice groups using the chosen parameter.

3) MBAmap design: the result of the classification process determines the macroblock to slice group map.

A great deal of research has been done to find the appropriate indicators to quantify the importance of a macroblock such as: bit-count [32], distortion-from-error concealment [33], distortion-from-propagation [34], a macroblock PSNR parameter [35], a macroblock impact factor [36], a macroblock importance factor [37] and mean square error (MSE) [38]. FMO can limit the error propagation within a slice and is useful for error concealment. The overhead required for FMO is acceptable. It has been observed that the total overhead incurred by omitting in-picture prediction is normally less than 10% [39].

SP and SI Slices

The switching P (SP) slice and the switching I (SI) slices are two special slices that enable efficient switching between video streams and efficient random access for video decoders [40]. For a video decoder, switching between one of several encoded streams is a common requirement in video streaming applications.

Figure 1.13 shows an example of Switching streams using I slices [41]. After decoding P slice A_0 and A_1 of stream A, the decoder wants to switch to stream B and decode B_2, B_3 and so on. The scheme is to code B_2 as I slice then the decoder can switch from stream A to stream B through B_2. It can also be realized by inserting an I slice at regular intervals in the coded sequence to create switching points, which has to increase the data rate as a cost.

SP-slices can realize switching between similar coded sequences without the increase of data rate of I slices. Figure 1.14 shows an example of switching streams using SP slices [41]. At the switching point, there are three SP slices coded using motion compensated prediction. SP slice A_2 can be decoded using reference picture A_1 and SP slice B_2 can be decoded using reference picture B_1. SP slice AB_2 is the switching point. It is created in such a way that it can be decoded using motion-compensated reference picture A_1 to produce B_2. In this way, the decoder can switch from stream A to stream B.

1.3.2 Scalable Video Coding

Scalable video coding is generated due to the demand of services to heterogeneous clients with various network conditions and device capabilities. In 2007, the Joint Video Team (JVT) approved a Scalable Video Coding (SVC) extension of the

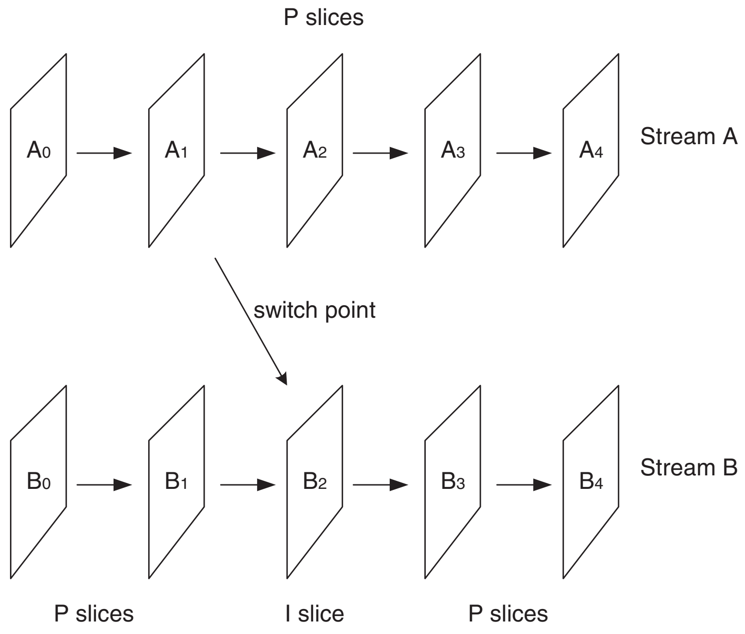


Fig. 1.13. Switching Streams Using I Slices.

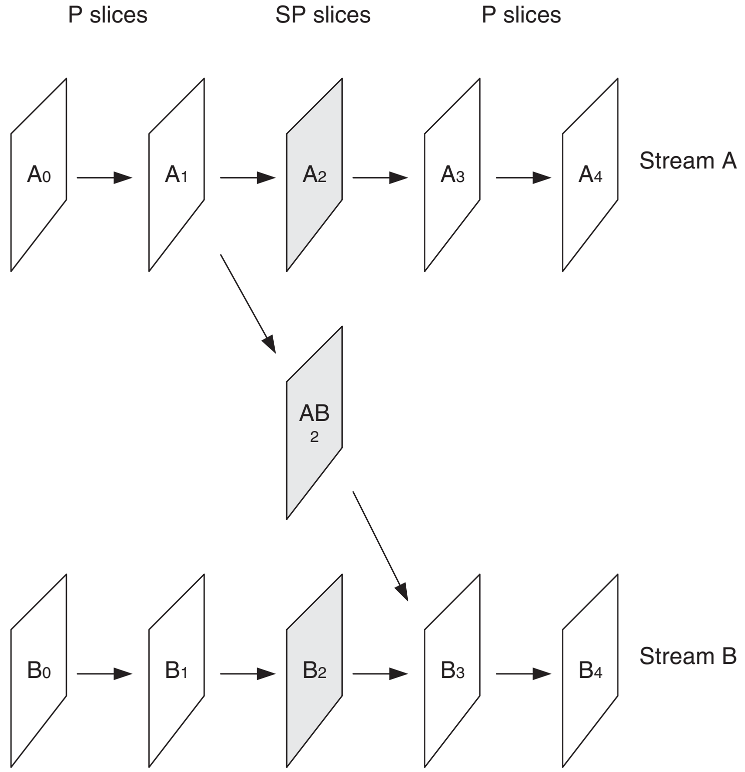


Fig. 1.14. Switching Streams Using SP Slices.

H.264/AVC standard, where significant improvement has been achieved over previous scalable schemes [42].

Figure 1.15 shows an example of the SVC encoder structure. In SVC, the original video data is partitioned into several hierarchical layers. The base layer contains the essential data and can be decoded at an acceptable quality. The enhancement layers contain additional refinement data for better decoding quality and are coded dependently. Each enhancement layer is coded based on all the layers below it in the hierarchy. Therefore, the base layer should be transmitted over more reliable channels. For error resilience purpose, the video source can discard a certain number of enhancement layers when network congestion occurs. In addition, unequal error protection can be used to scalable video coding since SVC has a hierarchy structure. For example, when forward error coding (FEC) is used for unequal error protection, the more important the layer is, the more parity bits are added to protect this layer.

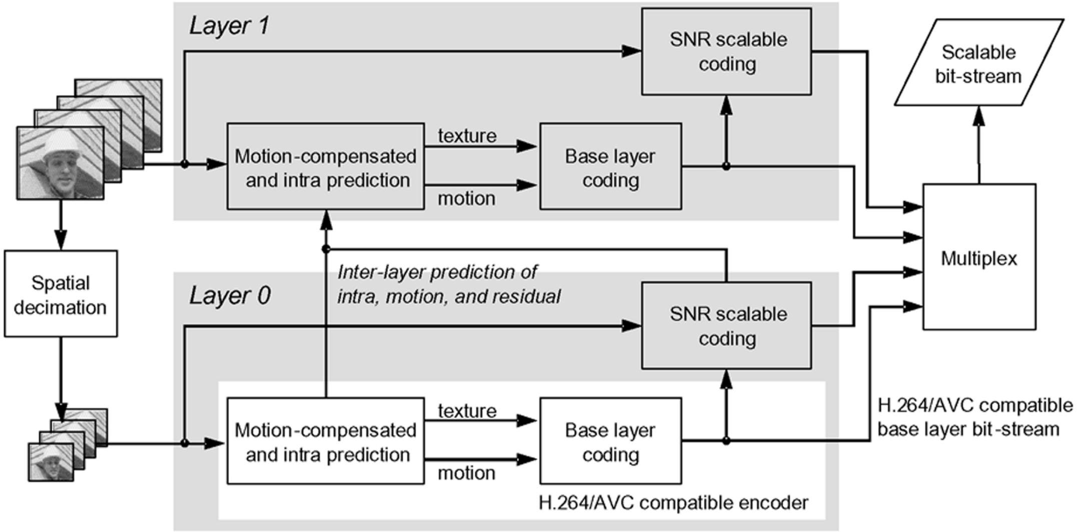

Layer 1
SNR scalable coding
Scalable bit-stream
texture
Motion-compensated and intra prediction
Base layer coding
motion
Multiplex
Spatial decimation
Inter-layer prediction of intra, motion, and residual
Layer 0
SNR scalable coding
texture
Motion-compensated and intra prediction
Base layer coding
motion
H.264/AVC compatible base layer bit-stream
H.264/AVC compatible encoder

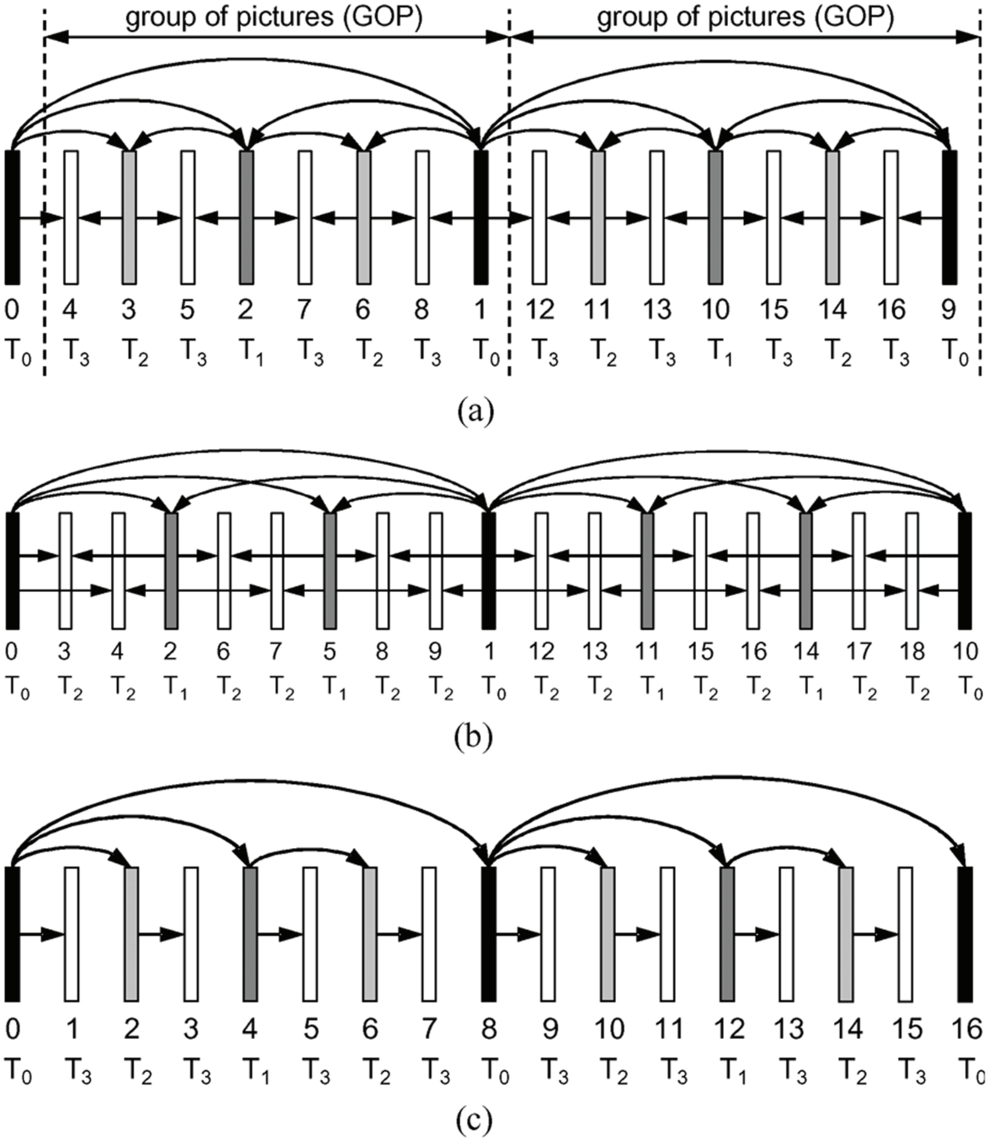
Fig. 1.15. An Example of the SVC Encoder Structure [42].

The usual modes of scalabilities in SVC are temporal, spatial and quality scalability [42].

Temporal scalability can be realized in a hierarchical structure as shown in Figure 1.16, which restricts motion-compensated prediction to reference pictures with a less or equal temporal layer identifier of the picture to be predicted. It usually improves the coding efficiency especially when a high delay can be tolerated.

In spatial scalability, the input video sequences for spatial layers are representations of the same video content with different resolutions. As shown in Figure 1.17, besides the prediction for single-layer coding, spatial scalable video coding also uses the inter-layer correlation to achieve coding efficiency higher than simulcast coding. The inter-layer prediction can use the information from lower layers as the reference. This ensures that a set of layers can be decoded independently of all higher layers. New pyramid and subband motion compensation methods are proposed in [43]. An efficient inter-layer motion-compensation technique is proposed for enhancement layer in [44–46]. The lower bound on the rate distortion performance of pyramid, subband and intra-layer motion compensation is studied in [47–49]. Other rate distortion analysis of scalable video coding is extensively studied in [50–52].

Coarse-grain quality scalability (CGS), medium-grain quality scalability (MGS) and fine-grain quality scalability (FGS) are three fundamental concepts in quality scalable coding. In CGS, quality scalability is achieved by using similar inter-layer prediction techniques as spatial scalability [53]. CGS is characterized by its simplicity in design and a low decoder complexity. However, it is lack of flexibility in the sense that switching between different CGS layers can only be done at defined points in the bitstream. To increase the flexibility of bitstream adaptation, medium-grain quality scalability (MGS) is introduced in the SVC standard. The concept of key pictures allows the adjustment of a suitable tradeoff between drift and enhancement layer coding efficiency for hierarchical predictions structures. For FGS coding in MPEG-4 Visual, the prediction structure is chosen in a way that a bitstream can be truncated and decoded at any point. Progressive refinement (PR) slices are used in FGS to achieve full quality scalability over a wide range of rate-distortion points. A motion compensation (MC) scheme for FGS scalable video coding is proposed in [54], where


group of pictures (GOP)
group of pictures (GOP)
0 4 3 5 2 7 6 8 1 12 11 13 10 15 14 16 9
T_0 T_3 T_2 T_3 T_1 T_3 T_2 T_3 T_0 T_3 T_2 T_3 T_1 T_3 T_2 T_3 T_0
0 3 4 2 6 7 5 8 9 1 12 13 11 15 16 14 17 18 10
T_0 T_2 T_2 T_1 T_2 T_2 T_1 T_2 T_2 T_0 T_2 T_2 T_1 T_2 T_2 T_1 T_2 T_2 T_0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
T_0 T_3 T_2 T_3 T_1 T_3 T_2 T_3 T_0 T_3 T_2 T_3 T_1 T_3 T_2 T_3 T_0

(a)

(b)

(c)

Fig. 1.16. Hierarchical Prediction Structures for Temporal Scalability. (a) Coding with hierarchical B-pictures. (b) Nondyadic hierarchical prediction structure. (c) Hierarchical prediction structure with a structural coding delay of zero. The numbers directly below the pictures specify the coding order, the symbols T_k specify the temporal layers with k representing the corresponding temporal layer identifier [42].

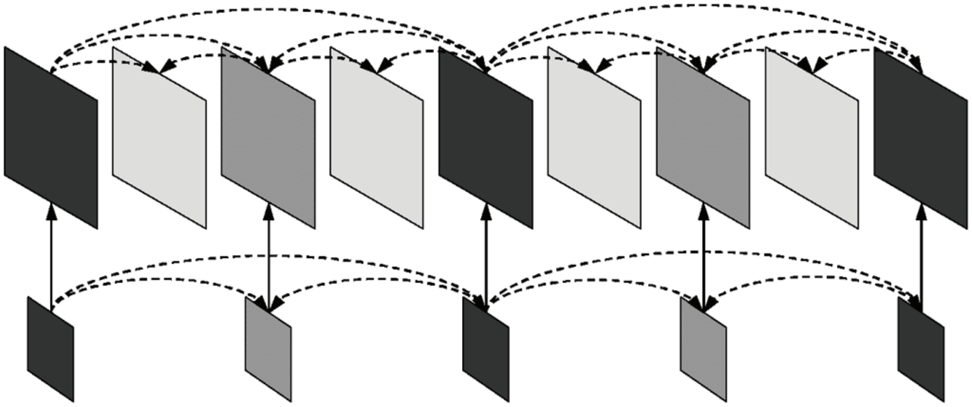


Fig. 1.17. Multilayer Structure with Additional Inter-Layer Prediction for Spatial Scalability [42].

two MC loops are used, one for the base layer and one for the enhancement layer. A technique called conditional replacement (CR), which adaptively selects between the base layer and enhancement layer prediction for each enhancement layer DCT coefficient, is used to simultaneously improve coding efficiency and reduce prediction drift.

In addition, because of the inherent scalability in wavelet transform, wavelet based video coding structure can be used for scalability. A fully rate scalable video codec, known as SAMCoW (Scalable Adaptive Motion Compensated Wavelet) is proposed in [55]. Many other approaches have been proposed in [56, 57].

1.3.3 Multiple Description Coding

Multiple description coding (MDC) has emerged as a promising approach to enhance the error resilience of a video delivery system [58]. In MDC, a single video source is partitioned into several equally important descriptions so that each description can be decoded independently at an acceptable decoding quality. The decoding quality is improved when more descriptions are received. A typical structure of MDC is depicted as Figure 1.18.

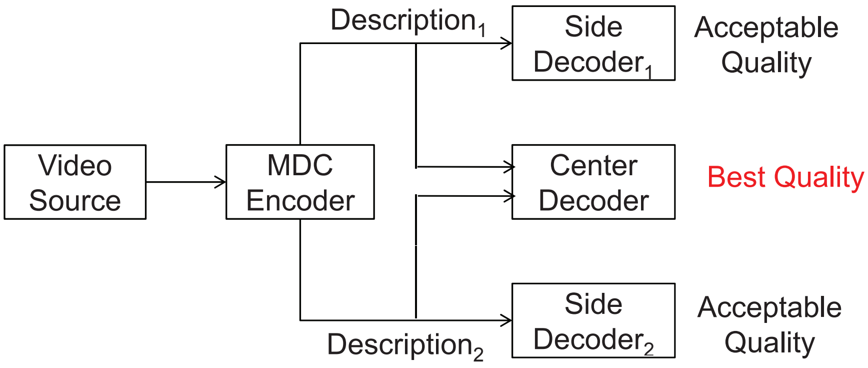


Fig. 1.18. Framework of MDC with Two Descriptions.

The encoded descriptions are sent through the same or different channels. When packet loss occurs, video bitstream is still decodable and any subset of the descriptions can reconstruct the original video with a reduced quality. Therefore, in cases of packet loss, MDC can provide an acceptable quality without retransmission. This advantage of MDC is very appealing to real-time interactive applications such as video conferencing, for which retransmission is often not acceptable. In contrast, scalable video coding (SVC), which is made up of a base layer and one or more enhancement layers, requires the base layer delivery almost error free. This leads to differential treatments of the base layer and enhancement layers. If packet loss occurs at the base layer, the video bitstream is not decodable for applications in which retransmission is not feasible.

A variety of MDC algorithms have been proposed in recent years. In [59], three classes have been defined based on the predictor type. Class A focuses on mismatch control: the partition is done before motion compensation so that each description is encoded independently, but this scheme reduces the prediction efficiency as a cost. Class B focuses on prediction efficiency: the partition is done after motion compensation so that the prediction efficiency is as good as single description coding. However, this scheme requires the reference frames to be fully reconstructed; otherwise mismatch can occur. Class C trades off between Class A and Class B. Moreover, to reduce mismatch, a compressed version of the mismatch or periodic I-frames can be used. More details about MDC are provided in Chapter 2.

1.3.4 Overview of The Thesis

Contributions of The Thesis

In this thesis, we investigated multiple description video coding with adaptive error concealment. The main contributions of the thesis are:

- Multiple Description Video Coding with Four Descriptions
  We describe a new four-description MDC method based on a hybrid structure of temporal and spatial correlations with spatial concealment as the default method and temporal concealment as a secondary method. Experimental results show that both scalability performance and packet loss performance work well.

- An Adaptive Error Concealment for MDC Based on Error Tracking
  We investigated an adaptable spatial-temporal error concealment method for multiple description video coding. This method takes into account the distortion due to error concealment and the distortion due to error propagation. Experimental results demonstrate that this method improves the non-adaptable method by adapting to any sequences.

- Macroblock-Level Adaptive Error Concealment Methods for MDC
  We investigated two adaptive spatial-temporal error concealment methods for MDC with four descriptions based on foreground-background mapping and distortion mapping respectively. Experimental results demonstrate that both methods improve the objective and subjective quality of the frame-level concealment based MDC.

- Advanced Multiple Description Video Coding for Better Coding Efficiency
  We proposed a new MDC partition architecture in which temporal separation

is outside the prediction loop and spatial separation is inside the loop. Experimental results show that this method improves the the objective quality but slightly degrades the subjective quality as a cost.

1.4 Publications Resulting from the Thesis

Journal Articles:

- **Meilin Yang**, Mary L. Comer, and Edward J. Delp, "Temporal-Spatial Based Multiple Description Video Coding Methods with Adaptive Error Concealment," *IEEE Transactions on Circuits and Systems for Video Technology*, in preparation.

Conference Papers

- Neeraj Gadgil, **Meilin Yang**, Mary L. Comer, and Edward J. Delp, "Adaptive Error Concealment for Multiple Description Video Coding Using Motion Vector Analysis," *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, Florida, U.S.A., September, 2012.

- **Meilin Yang**, Mary L. Comer, and Edward J. Delp, "Macroblock-Level Adaptive Error Concealment Methods for MDC," *Proceedings of the Picture Coding Symposium (PCS)*, Krakow, Poland, May, 2012.

- **Meilin Yang**, Ye He, Fengqing Zhu, Marc Bosch, Mary L. Comer, and Edward J. Delp, "Video Coding: Death is Not Near," *Proceedings of the 53rd International Symposium ELMAR*, Zadar, Croatia, September 2011, pp. 85-88.

- **Meilin Yang**, Mary L. Comer, and Edward J. Delp, "An Adaptable Spatial-Temporal Error Concealment Method for Multiple Description Coding Based on Error Tracking," *Proceedings of IEEE International Conference on Image Processing (ICIP)*, Brussels, Belgium, September 2011, pp. 337-340.

- Rohan Gandhi, **Meilin Yang**, Dimitrios Koutsonikolas, Yu Charlie Hu, Mary Comer, Amr Mohamed, and Chih-Chun Wang, "The Impact of Inter-layer Network Coding on the Relative Performance of MRC/MDC WiFi Media Delivery," *Proceedings of the 21th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, Vancouver, Canada, June, 2011, pp. 27-32.

- **Meilin Yang**, Mary L. Comer, and Edward J. Delp, "A Four-Description MDC for High Loss-Rate Channels," *Proceedings of the Picture Coding Symposium (PCS)*, Nagoya, Japan, December 2010, pp. 418-421.

- Nitin Khanna, Fengqing Zhu, Marc Bosch, **Meilin Yang**, Mary L. Comer, and Edward J. Delp, "Information Theory Inspired Video Coding Methods: Truth Is Sometimes Better Than Fiction", *Proceedings of the Third Workshop on Information Theoretic Methods in Science and Engineering*, Tampere, Finland, August, 2010.

2. OVERVIEW OF MULTIPLE DESCRIPTION VIDEO CODING

With the popularity of real-time video streaming, such as IPTV and peer-to-peer live video, it is important to consider bandwidth fluctuation, heterogeneous clients and high packet loss rate transmission problems. Multiple Description Coding (MDC) has been proposed as an effective solution to combat error-prone channels, particularly for real-time applications when retransmission is unacceptable. MDC generates two or more equally important "descriptions" of the video sequence. Each description can be decoded independently at an acceptable quality. The decoding quality is improved when more descriptions are received. A typical structure of MDC is shown in Figure 2.1. If one description is received, the side decoder reconstructs the signal with distortion D_1 or D_2. If both descriptions are received, the central decoder reconstructs the signal with distortion D_0, and $D_0 \leq \min\{D_1, D_2\}$. It is not possible to simultaneously minimize both D_0 and $(D_1 + D_2)$ [59]. In fact, the performance achieved by using all the descriptions at the decoder can be obtained by the corresponding single description coder (SDC) with a smaller total rate [60]. The redundancy introduced in MDC is to improve the error robustness, and a main challenge for MDC is to control the redundancy.

The pioneering MD video coder known as a multiple description scalar quantizer (MDSQ) [61] was proposed in 1993. In this approach, two sub-streams are obtained by using two indexes at different quantization levels. The index assignment is designed in a way that if both indexes are received, the reconstruction is equivalent to using a fine quantizer while if only one index is received, the reconstruction is equivalent to using a coarse quantizer. One way to design the indexes is use two quantizers whose decision regions are shifted by half of the quantizer interval with respect to each other [25]. If

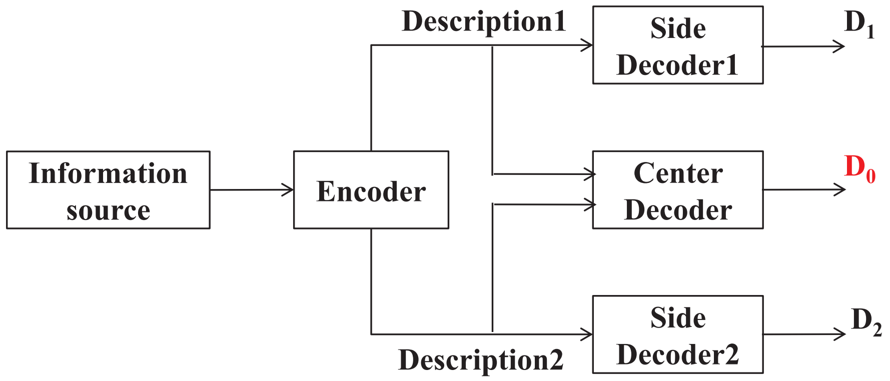


Fig. 2.1. The MDC Architecture.

each quantizer need data rate R, then the total data rate is $2R$. The side distortion is equivalent to a single quantizer with data rate R, and the central distortion is equivalent to a single quantizer with data rate $R+1$. Therefore, the coding efficiency is significantly degraded at high data rate. In this approach, the index streams are assumed to be coded using fixed length codes. To improve the coding efficiency, the index streams are assumed to be coded using variable length codes, and the MDSQ approach is extended to entropy-constrained multiple description scalar quantizers (ECMDSQs) [62]. The primary MDSQ was developed for memoryless sources, and an asymptotic analysis has been presented in [63]. For sources with memory such as communicating over a Rayleigh fading channel and for communicating over a lossy packet network, a multiple description transform coder (MDTC) is proposed and an asymptotic analysis is presented in [64, 65].

Another popular MDC approach exploits a pairwise correlating transform (PCT) to introduce dependencies between two descriptions [66–68]. The general idea is to divide the transform coefficients into several groups so that the coefficients between different groups are correlated while the coefficients within the same group are uncorrelated. In this way, the lost coefficient groups can be concealed by the received groups and by decorrelating the coefficients within the same group, coding efficiency is maximized. Instead of using the Karhunen-Loeve transform (KLT) that decorrelates all the coefficients, the authors design the transform bases in a way that the coefficients are correlated pair-wise. In [66], a pairwise correlating transform is used on each pair of uncorrelated coefficients generated by KLT. This is realized by rotating every two basis vectors in the KLT with 45 degree. Each pair of correlated coefficients are then split into two descriptions with equal variance and are coded independently at the same data rate. If both descriptions are received, an inverse PCT is used to reconstruct the coefficients. If only one description is received, the lost description can be concealed based on the correlation between the two descriptions. More general pairwise transforms are proposed in [67] including both orthogonal approaches and non-orthogonal approaches. The overhead can be controlled by the number of paired coefficients and the transform parameters. This approach has been used for motion compensated multiple description video coding in [69].

Instead of designing the transform basis to introduce correlation among coefficients within the same block, an alternative approach is to introduce correlation among the same coefficients in different blocks. Besides, to introduce additional correlation, overlapping block transforms can be used. In [70], lapped orthogonal transform (LOT) bases are proposed, which can achieve tradeoffs between coding efficiency and error robustness. In this method, a missing coefficient block is concealed by the average of its available neighbors. A transform is selected based on the channel characteristics, the desired reconstruction performance, and the desired coding efficiency. A LOT-DCT basis is proposed in [71], which is designed to maximize the coding efficiency, but the complexity is increased by 20-30 percent.

The methods mentioned above, although providing good performance, generate descriptions that are not compatible with video standards such as H.264 [7]. This leads to the use of MDC at pre- or/and post-processing stages [72]. The general idea is to split the video source into two sub-videos, which are encoded independently. At the decoder side, when the two descriptions are received, the decoded sub-videos are postprocessed to recover the full quality video. If one description is lost, the received one can reconstruct the video at a coarser quality. In [73], the authors use a oversampling method to add redundancy to the original image data followed by a partitioning of the oversampled image into equal sub-images. It has been extended to video applications in [74]. In these methods, zero padding is used in two-dimensional DCT domain. In [75], an arbitrary number of descriptions, based on zero padding in the DCT domain followed by MD generation using polyphase downsampling is described. A simple way to generate two description is using horizontal or vertical downsampling. To generate more than two descriptions, the partition is in a way that redundancy is uniformly distributed along the image columns and rows. This scheme does not represent the best partition to generate multiple descriptions, because different correlations can be distributed along image columns and rows. Similar ideas can be found in [76, 77]. To protect a region of interest, a content-based multiple description image coding is proposed in [78]. A nonlinear geometrical transform is used add redundancy mainly to the region of interest followed by splitting the transformed image into sub-images which are coded and transmitted separately. Other downsampling-based MDC in the spatial, temporal or frequency domain can be found in [79–85].

Goyal [60] and Wang [59] have presented comprehensive reviews of MDC methods. In [59], three classes have been defined based on the predictor type. Class A focuses on mismatch control: MDC is done before motion compensation so that each description can be encoded independently, but this reduces prediction efficiency. “Mismatch” means the reconstructed frame at the decoder is different from the one at the encoder due to packet loss. In [80], the authors use an simple approach to splits the video sequence into odd and even frames, and separately encodes the two groups to form two independent descriptions. Similar approaches can be found in [77] and [86]. Class B focuses on prediction efficiency: MDC is done after motion compensation so that the prediction efficiency is as good as single description coding (SDC). But this method requires the reference frames to be fully reconstructed, otherwise mismatch can be induced. In [87], the authors propose the MD-split method, in which motion vectors and low-frequency coefficients are alternated between the descriptions. Another example can be found in [88]. Class C trades off between Class A and Class B. In [89], the authors propose multiple description motion compensation (MDMC) method, where the central predictor forms a linear superposition of the past two reconstructed frames, and the superposition weights control both redundancy and mismatch. Moreover, to reduce mismatch, a compressed version of the mismatch or periodic I-frames can be used.

To analyze these various MDC approaches in a more straightforward way, another classification is proposed in [90]. Since the video source can be partitioned into multiple descriptions at different stages during encoding such as preprocessing, encoding and postprocessing stage, the MDC methods can be classified according to at which stage the partition is done. For MDC methods with descriptions generated at preprocessing stage, the original video is split into multiple sub-videos before encoding and these sub-videos are encoded independently to generate multiple descriptions. Typical examples are downsampling-based MDC in the spatial and temporal domain as have been described. For MDC methods with descriptions generated at encoding stage, the one-to-multiple mapping is realized by particular coding techniques such as MDSQ, PCT-based MDC and LOT-DCT based MDC as have been discussed. For MDC methods with descriptions generated at postprocessing stage, the one-to-multiple mapping is realized by splitting an encoded bitstream to multiple sub-streams in the compressed domain. A FEC-based MDC is proposed in [91]. In this method, maximum distance separable (n, k) erasure channel codes are used to generate multiple sub-streams using a joint source-channel coding method. And the redundancy can be controlled by k. Each class in these three categories has advantages and disadvantages. And a combination of different categories could achieve a better performance according to the specific application. The major advantage of preprocessing-stage MDC methods is that the coding scheme remains unchanged. Thus the generated descriptions are compatible with all the existing video standards, which is convenient to implement. MDC methods based on preprocessing scheme take advantage of the assumption that spatially or temporally adjacent video data samples are highly correlated. Therefore, video data samples in one description can be well estimated from the corresponding data samples in other descriptions. One simple way to implement the preprocessing data partition is to split odd and even frames of a video sequence into two sub-sequences. Each of the newly generated sub-sequences is encoded using a video coding standard such as H.264. Some standard-compliant downsampling-based MDC methods in temporal domain were described [80, 81]. Another simple way to implement the preprocessing data partition is to split odd and even columns of each frame in a video sequence into two sub-sequences. Similarly, each of the newly generated sub-sequences is encoded using a video coding standard. An example of the spatial downsampling-based MDC methods can be found in [92]. Since the correlations among neighboring pixels are reduced after downsampling, coding efficiency is degraded. MDC algorithms based on frequency-domain downsampling can alleviate this correlation-reduction problem. One approach to implement the preprocessing data partition in the frequency domain is to add zeros to the transform coefficients in both dimensions after DCT, followed by MD generation using polyphase downsampling [74]. Another approach to implement the domain-based MDC is to split the wavelet coefficients into maximally separated sets [82], which can use simple error concealment methods to produce good estimates of lost signal samples. However, these two schemes fail to exploit temporal correlations very well. To combine the advantages of scalable coding and MDC, a multiple description scalable video coding (MDSVC) scheme based on motion-compensated temporal filtering (MCTF) [93] is described in [83]. Video frames are filtered into low-frequency and high-frequency frames, prior to the spatial transformation and coding, where high-frequency frames are separated into two descriptions and the lost frames can be estimated from low-frequency frames along with the motion vector information. In [84], the authors describe a MDC method based on fully scalable wavelet video coding, in which post encoding is done to adapt the number of descriptions, the redundancy level and the target data rate.

Due to the demand of applications in scalable, multicast and P2P environments, there has been an increasing interest in MDC methods with more than two descriptions. Zhu and Liu [90] propose a multi-description video coding based on hierarchical B pictures where temporal-level-based key pictures are selected in a staggered way among different descriptions. In [90], the authors use a linear combination of received descriptions to optimize decoding results. However, this approach suffers from high data-rate redundancy by duplicating the original sequence into two descriptions. Hsiao and Tsai [94] present a four-description MDC which takes advantage of residual-pixel correlation in the spatial domain and coefficient correlation in the frequency domain. However, this method fails to exploit the temporal correlation and mismatch is prone to be induced.

In this dissertation, we first propose a MDC partition architecture with four descriptions based on temporal and spatial correlations. In this architecture, the MDC partition is done before the prediction so that each description has an independent prediction loop. This method has good mismatch control. To improve the error robustness, we develop several adaptive error concealment methods. One method is frame level based on error tracking. It takes into account the distortion due to error concealment and error propagation. Other methods are on the macroblock level based on foreground-background mapping and distortion mapping respectively. We also propose a MDC partition architecture with better coding efficiency. In this method, a spatial partition is done after the prediction so that each of the two descriptions share the same prediction loop. This method has a better PSNR performance but the visual quality is slight worse than the first proposed architecture.

3. A TEMPORAL-SPATIAL BASED MDC WITH FOUR DESCRIPTIONS

Generally, MDC generates two equally important descriptions so that each description can be decoded independently at an acceptable quality. Moreover, the decoding quality is improved when both descriptions are received. For applications in scalable, multicast and P2P environments, it is advantageous to use more than two descriptions. In this chapter, we propose a MDC method with four descriptions using both temporal and spatial correlations. This chapter is organized as follows: Section 3.1 provides a detailed description of the proposed MDC architecture. Section 3.2 describes the error concealment schemes for the proposed MDC method. Section 3.3 introduces the Gilbert model for packet loss simulation. Experimental results and discussions are provided in Section 3.4. And we conclude the chapter in Section 3.5.

3.1 Framework of the Proposed Method

Figure 3.1 shows our proposed MDC architecture [95–97]. The original video sequence is first split into two temporally correlated descriptions, which are known as the Even and Odd sequences respectively. Each of the new sequences is then separated into two spatially correlated sequences through horizontal downsampling. As a result, Even_1 and Odd_1 are from odd columns of each frame, while Even_2 and Odd_2 are from even columns. “Even” and “Odd” denotes temporally correlated even numbered frames and odd numbered frames. “1” and “2” denotes spatially correlated odd columns and even columns of each frame. For example, “Even_1” denotes the odd columns from even-numbered frames after horizontal downsampling. The newly generated four descriptions are encoded independently and are sent through the same or different channels. This is a Class A method as we discussed in the previous chapter [59], which has good mismatch control but loses some coding efficiency.

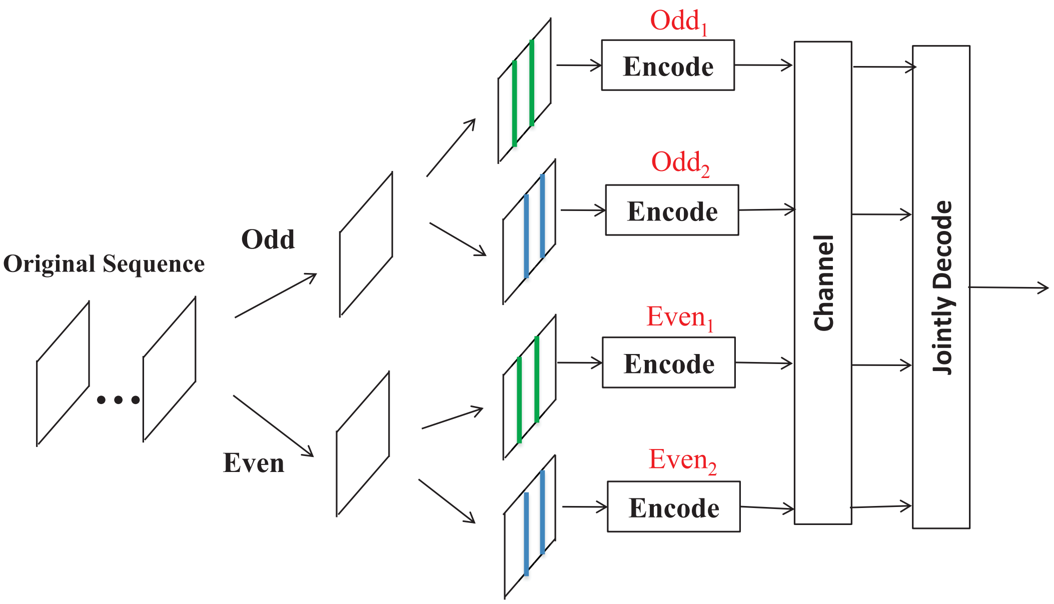


Fig. 3.1. Framework of the Proposed Method.

3.2 Error Concealment Schemes

When the four descriptions, defined in the previous subsection, are correctly received, each description can be decoded independently. The final reconstructed video is the combination of the four decoded sub-videos. However, when packet loss occurs during transmission, joint decoding is done.

Table 3.1 shows the error concealment scheme for each packet loss scenario, where the first row indicates which descriptions of the Odd sequence are received and the first column indicates which descriptions of the Even sequence are received. “Odd$_1$+Odd$_2$” means both Odd$_1$ and Odd$_2$ sequences are received, “Odd$_1$” means only Odd$_1$ of the Odd sequence is received, and “Loss” in the first row denotes that neither Odd$_1$ nor Odd$_2$ is received. Similarly, “Even$_1$+Even$_2$” denotes both Even$_1$ and Even$_2$ sequences are received, “Even$_1$” means only Even$_1$ of the Even sequence is received, and “Loss”

Concealment Methods	Odd_1+Odd_2	Odd_1	Odd_2	Loss
$Even_1$+$Even_2$	N/A	Spatial	Spatial	Temporal
$Even_1$	Spatial	Spatial	Spatial	Spatial-Temporal
$Even_2$	Spatial	Spatial	Spatial	Spatial-Temporal
Loss	Temporal	Spatial-Temporal	Spatial-Temporal	Frame Repeat

Table 3.1
Error Concealment Schemes.

in the first column denotes neither Even_1 nor Even_2 is received. “Spatial-Temporal” means spatial concealment is done first then temporal concealment is done. According to Table 3.1, when one description is received, such as Odd_1, we use spatial concealment to conceal Odd_2 first and then use temporal concealment to conceal Even_1 and Even_2. When two descriptions are received, if the two descriptions are from the same spatial correlation such as Odd_1 and Odd_2, we use temporal concealment to conceal Even_1 and Even_2, otherwise we use spatial concealment. When three descriptions are received, we use spatial concealment. When the four descriptions are lost at the same time, we use the previously decoded reference frame for concealment which is called “Frame Repeat” in the table. In conclusion, we use spatial concealment as the default method and temporal concealment as the secondary method.

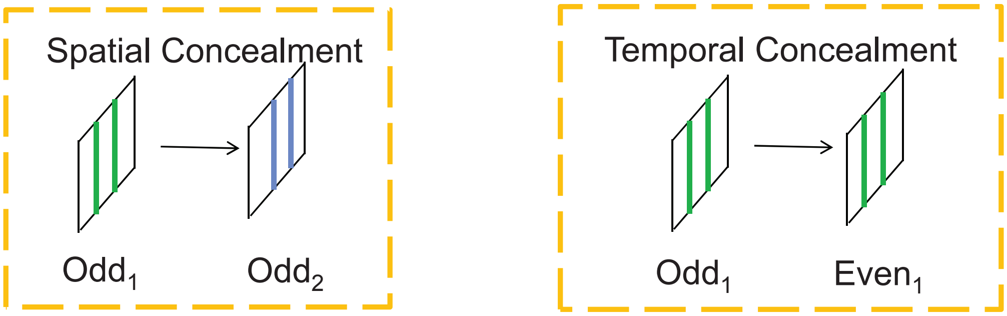


Fig. 3.2. Spatial and Temporal Correlation for Error Concealment.

Figure 3.2 illustrates the spatial and temporal correlation for error concealment. For spatial concealment, we use a two-neighbor bilinear filter. For temporal concealment, we use a non-motion compensated method which copies the pixel value of the same position from the frame used for concealment.

3.3 Channel Model

Packet loss patterns in a typical network are usually bursty. Therefore, simply using symmetric packet loss model can not represent realistic network transmission characteristics. In this thesis, a Gilbert model is used as the channel model for burst packet loss [98].

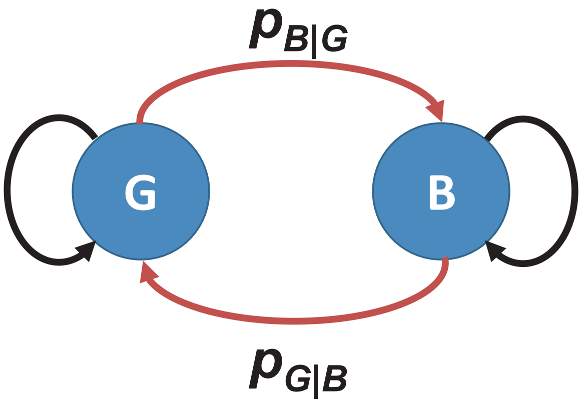


Fig. 3.3. Gilbert Model (Two State Markov Model).

Figure 3.3 illustrates the architecture of the Gilbert model. In this two-state model, condition "good" denoted as "G" indicates that the packet is correctly received and condition "bad" denoted as "B" indicates that the packet is lost. The transition probability $P_{M|N}$ is the probability of changing the state from N to M. For example, $P_{B|G}$ is the probability when the previous state is "G" and the current state is "B", and $P_{G|G}$ is the probability when the previous state is "G" and the current state is also "G". Therefore:

$$\begin{aligned} P_{G|G} &= 1 - P_{B|G} \\ P_{B|B} &= 1 - P_{G|B} \end{aligned} \tag{3.1}$$

The probability of packet loss can be expressed as:

$$\begin{aligned} P_B &= P_{B|B} \times P_B + P_{B|G} \times P_G \\ &= (1 - P_{G|B}) \times P_B + P_{B|G} \times (1 - P_B) \end{aligned} \tag{3.2}$$

After reorganization, we have:

$$P_B = \frac{P_{B|G}}{P_{B|G} + P_{G|B}} \tag{3.3}$$

The packet loss burst length is the average length of "consecutive stays" in the state "B:"

$$\begin{aligned} L_B &= 1 \times P_{G|B} + 2 \times P_{B|B} \times P_{G|B} + 3 \times P_{B|B}^2 \times P_{G|B} + ... \\ &= P_{G|B}^{-1} \end{aligned} \tag{3.4}$$

In [99] a typical packet loss rate P_B is between 0 and 0.6, and burst length L_B is between 2 and 20. It can also be shown that when P_B is small, L_B is large; and vice versa.

3.4 Experimental Results

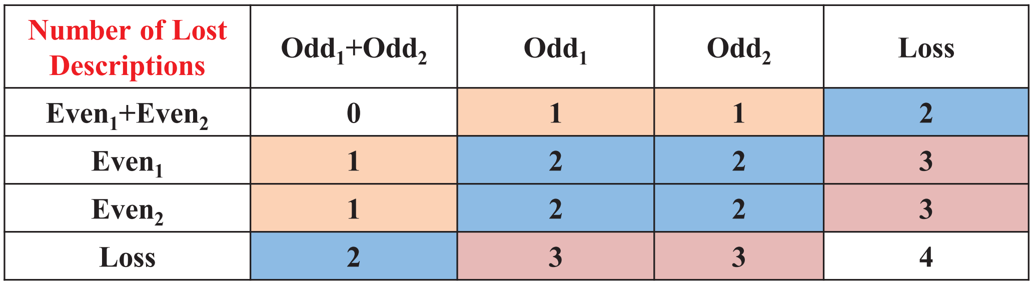
In this section, we evaluate the proposed method from two aspects: scalability performance and packet loss performance.

The experiments are implemented by modifying the JVT JM software version 16.1. The testing sequences have original resolutions of CIF (352 × 288) at 30 frames/sec with 200-frame length. Thus each description has 100 subframes at 15 frames/sec. The coding structure is "IPPP...", with an I-frame refreshment every 30 subframes in each description. The quantization parameters for I frames and P frames are 18, 22, 26 and 30.

Nine testing sequences are used in our experiments: *"Bridge"*, *"Akiyo"*, *"Mother"*, *"Flower"*, *"Foreman"*, *"Crew"*, *"Ice"*, *"Football"* and *"Soccer"*. Among these testing sequences, *"Bridge"* and *"Akiyo"* are low-motion sequences. *"Mother"* and *"Flower"* are low-to-medium-motion sequences. *"Foreman"*, *"Crew"* and *"Ice"* are medium-to-high-motion sequences. *"Football"* and *"Soccer"* are high-motion sequences. Experimental results are obtained by averaging 100 channel transmission simulations using the Gilbert model described above.

3.4.1 Scalability Performance

At the decoder, the client can decide how many descriptions it wants to receive according to its bandwidth. Scalability performance is evaluated when some of the four descriptions are totally lost, and the others are correctly received. Table 3.2 shows the number of lost descriptions for each scenario, where the first row lists scenarios for Odd frames and the first column lists scenarios for Even frames. There are sometimes more than one scenario when a number of descriptions are lost. For example, there are 4 scenarios when one description is lost, 6 scenarios when two descriptions are lost, and 4 scenarios when three descriptions are lost. Therefore, to measure the scalability performance, the results are obtained by averaging all the scenarios.



Number of Lost Descriptions	Odd_1+Odd_2	Odd_1	Odd_2	Loss
$Even_1$+$Even_2$	0	1	1	2
$Even_1$	1	2	2	3
$Even_2$	1	2	2	3
Loss	2	3	3	4

Table 3.2
Number of Lost Descriptions for Each Scenario.

Table 3.3 to Table 3.11 show the PSNR and data rate for each receiving scenario of the testing sequences. According to Table 3.2, there are 6 scenarios when two descriptions are received. As indicated in Table 3.1, when both the spatially correlated descriptions of Odd frames or Even frames are lost, temporal correlation is used for error concealment; otherwise, spatial correlation is always used to recover the lost data. Therefore, when Odd_1 and Odd_2 are received or Even_1 and Even_2 are received, temporal concealment is used. For the other 4 out of the 6 scenarios, spatial concealment is used. Table 3.3 to Table 3.6 show the results for the *"Bridge*, *"Akiyo"*, *"Mother"* and *"Flower"* sequence respectively. It is illustrated that for these sequences, temporal concealment performs better than spatial concealment. Table 3.7 to Table 3.9 shows the results for the *"Foreman"*, *"Crew"* and *"Ice"* sequence respectively. For these sequences, temporal concealment performs in a similar way as spatial concealment. Table 3.10 and Table 3.11 show the results for the *"Football"* and *"Soccer"* sequence respectively where spatial concealment performs better than temporal concealment. Therefore, it seems that temporal concealment works better for low-to-medium-motion sequences, and spatial concealment works better for medium-to-high-motion sequences.

Figure 3.4 illustrates the scalability performance of the proposed method. The horizontal axis denotes the number of descriptions that are received and the vertical axis is the average luma PSNR. The proposed method provides a graceful degradation with decreasing of the number of received descriptions. Moreover, even when only one description is received, the proposed method can still achieve an acceptable quality.

It is also shown that the performance for the *"Bridge"*, *"Akiyo"*, *"Mother"* and *"Flower"* sequence are nearly convex. The performance for the *"Foreman"*, *"Crew"* and *"Ice"* sequence are approximately a linear function. And the performance for the *"Football"* and *"Soccer"* sequence are concave. Apparently, the scalability performance in a convex curve degrades the fastest and the scalability performance in a concave curve degrades the slowest. Therefore, it seems that the scalability performance of the medium-to-high-motion sequences are better than the performance of the low-to-medium-motion sequences, which is consistent with what we have described in Table 3.3 to Table 3.11.

Table 3.3
Scalability Performance for the *"Bridge"* Sequence.

Received Descriptions	PSNR (dB)	Data Rate (kbps)
$odd_1 + odd_2 + even_1 + even_2$	44.56	4332.69
$odd_2 + even_1 + even_2$	38.57	3250.20
$odd_1 + even_1 + even_2$	38.65	3251.31
$odd_1 + odd_2 + even_2$	38.56	3247.28
$odd_1 + odd_2 + even_1$	38.65	3249.28
$odd_1 + odd_2$	39.77	2163.87
$even_1 + even_2$	39.86	2168.82
$odd_1 + even_1$	32.74	2167.9
$odd_2 + even_2$	32.57	2164.79
$odd_1 + even_2$	32.66	2165.9
$odd_2 + even_1$	32.65	2166.79
odd_1	32.10	1082.49
odd_2	31.95	1081.38
$even_1$	31.98	1085.41
$even_2$	32.04	1083.41

Table 3.4
Scalability Performance for the *"Akiyo"* Sequence.

Received Descriptions	PSNR (dB)	Data Rate (kbps)
$odd_1 + odd_2 + even_1 + even_2$	45.65	842.71
$odd_2 + even_1 + even_2$	40.37	631.89
$odd_1 + even_1 + even_2$	40.20	631.19
$odd_1 + odd_2 + even_2$	40.37	632.41
$odd_1 + odd_2 + even_1$	40.20	632.64
$odd_1 + odd_2$	42.76	422.34
$even_1 + even_2$	42.87	420.37
$odd_1 + even_1$	34.75	421.12
$odd_2 + even_2$	35.09	421.59
$odd_1 + even_2$	34.92	420.89
$odd_2 + even_1$	34.92	421.82
odd_1	34.29	210.82
odd_2	34.60	211.52
$even_1$	34.31	210.30
$even_2$	34.62	210.07

Table 3.5
Scalability Performance for the *"Mother"* Sequence.

Received Descriptions	PSNR (dB)	Data Rate (kbps)
$odd_1 + odd_2 + even_1 + even_2$	45.02	1647.95
$odd_2 + even_1 + even_2$	41.77	1236.89
$odd_1 + even_1 + even_2$	41.95	1236.42
$odd_1 + odd_2 + even_2$	41.77	1235.80
$odd_1 + odd_2 + even_1$	41.95	1234.74
$odd_1 + odd_2$	40.40	822.59
$even_1 + even_2$	40.45	825.36
$odd_1 + even_1$	38.88	823.21
$odd_2 + even_2$	38.52	824.74
$odd_1 + even_2$	38.70	824.27
$odd_2 + even_1$	38.70	823.68
odd_1	36.77	411.06
odd_2	36.51	411.53
$even_1$	36.79	412.15
$even_2$	36.52	413.21

Table 3.6
Scalability Performance for the *"Flower"* Sequence.

Received Descriptions	PSNR (dB)	Data Rate (kbps)
$odd_1 + odd_2 + even_1 + even_2$	45.79	6965.81
$odd_2 + even_1 + even_2$	36.39	5221.50
$odd_1 + even_1 + even_2$	36.42	5226.03
$odd_1 + odd_2 + even_2$	36.38	5223.75
$odd_1 + odd_2 + even_1$	36.43	5226.15
$odd_1 + odd_2$	31.78	3484.09
$even_1 + even_2$	31.78	3481.72
$odd_1 + even_1$	27.07	3486.37
$odd_2 + even_2$	26.99	3479.44
$odd_1 + even_2$	27.02	3483.97
$odd_2 + even_1$	27.03	3481.94
odd_1	22.87	1744.31
odd_2	22.83	1739.78
$even_1$	22.88	1742.06
$even_2$	22.82	1739.66

Table 3.7
Scalability Performance for the *"Foreman"* Sequence.

Received Descriptions	PSNR (dB)	Data Rate (kbps)
$odd_1 + odd_2 + even_1 + even_2$	44.22	3860.25
$odd_2 + even_1 + even_2$	40.16	2902.54
$odd_1 + even_1 + even_2$	39.92	2892.02
$odd_1 + odd_2 + even_2$	40.16	2898.64
$odd_1 + odd_2 + even_1$	39.91	2887.55
$odd_1 + odd_2$	34.97	1925.94
$even_1 + even_2$	34.98	1934.31
$odd_1 + even_1$	35.61	1919.32
$odd_2 + even_2$	36.11	1940.93
$odd_1 + even_2$	35.86	1930.41
$odd_2 + even_1$	35.86	1929.84
odd_1	30.60	957.71
odd_2	30.87	968.23
$even_1$	30.61	961.61
$even_2$	30.89	972.70

Table 3.8
Scalability Performance for the *"Crew"* Sequence.

Received Descriptions	PSNR (dB)	Data Rate (kbps)
$odd_1 + odd_2 + even_1 + even_2$	44.76	4698.11
$odd_2 + even_1 + even_2$	39.88	3525.73
$odd_1 + even_1 + even_2$	39.87	3528.67
$odd_1 + odd_2 + even_2$	39.87	3519.02
$odd_1 + odd_2 + even_1$	39.87	3520.91
$odd_1 + odd_2$	35.19	2341.82
$even_1 + even_2$	35.20	2356.29
$odd_1 + even_1$	34.98	2351.47
$odd_2 + even_2$	35.00	2346.64
$odd_1 + even_2$	34.98	2349.58
$odd_2 + even_1$	34.99	2348.53
odd_1	30.33	1172.38
odd_2	30.35	1169.44
$even_1$	30.33	1179.09
$even_2$	30.34	1177.20

Table 3.9
Scalability Performance for the *"Ice"* Sequence.

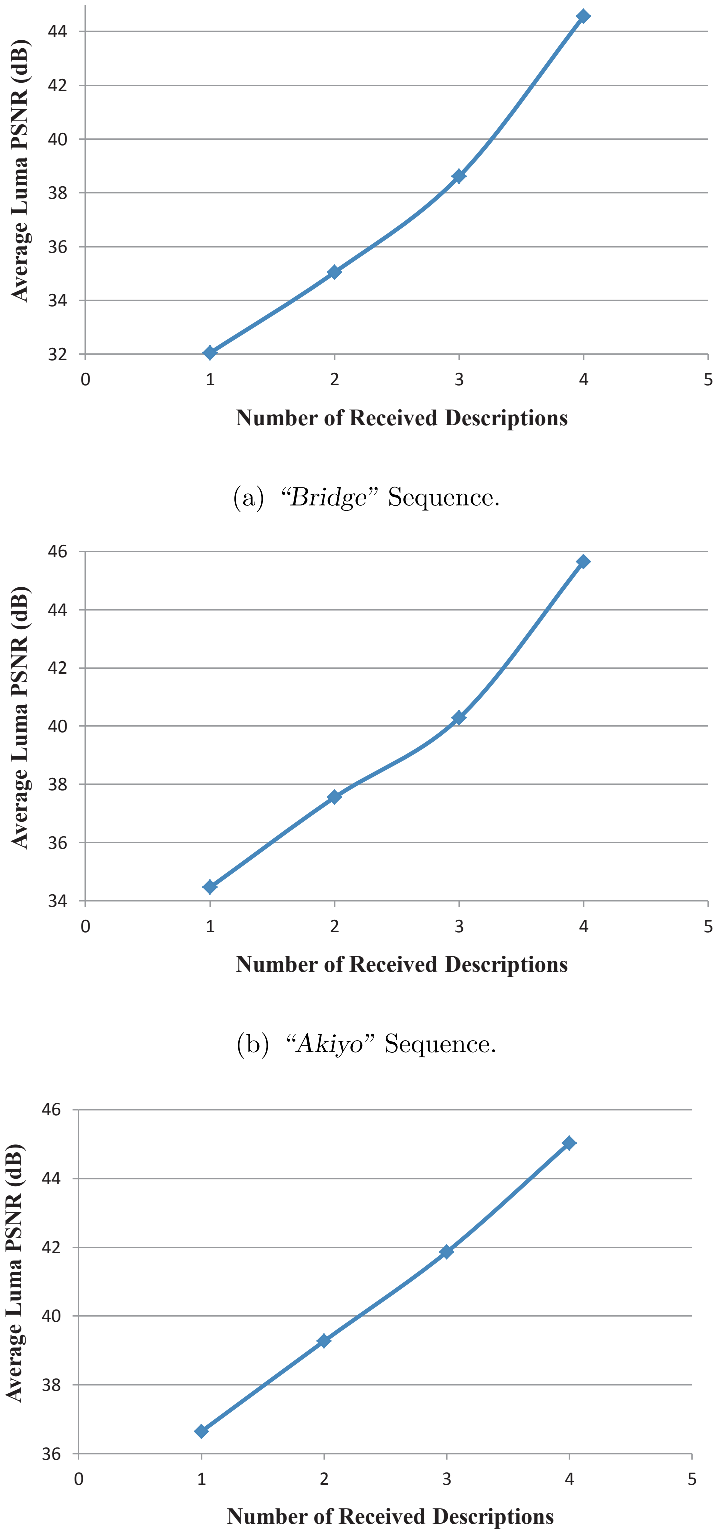
Received Descriptions	PSNR (dB)	Data Rate (kbps)
$odd_1 + odd_2 + even_1 + even_2$	46.32	2152.96
$odd_2 + even_1 + even_2$	40.97	1613.88
$odd_1 + even_1 + even_2$	41.01	1614.56
$odd_1 + odd_2 + even_2$	41.00	1615.26
$odd_1 + odd_2 + even_1$	41.01	1615.18
$odd_1 + odd_2$	34.83	1077.48
$even_1 + even_2$	34.84	1075.48
$odd_1 + even_1$	35.70	1076.78
$odd_2 + even_2$	35.66	1076.18
$odd_1 + even_2$	35.69	1076.86
$odd_2 + even_1$	35.67	1076.10
odd_1	29.64	539.08
odd_2	29.60	538.40
$even_1$	29.63	537.70
$even_2$	29.62	537.78

Table 3.10
Scalability Performance for the *"Football"* Sequence.

Received Descriptions	PSNR (dB)	Data Rate (kbps)
$odd_1 + odd_2 + even_1 + even_2$	44.75	5430.33
$odd_2 + even_1 + even_2$	39.90	4066.10
$odd_1 + even_1 + even_2$	40.62	4079.28
$odd_1 + odd_2 + even_2$	39.89	4067.27
$odd_1 + odd_2 + even_1$	40.61	4078.34
$odd_1 + odd_2$	32.11	2715.28
$even_1 + even_2$	32.11	2715.05
$odd_1 + even_1$	36.48	2727.29
$odd_2 + even_2$	35.03	2703.04
$odd_1 + even_2$	35.76	2716.22
$odd_2 + even_1$	35.75	2714.11
odd_1	28.06	1364.23
odd_2	27.32	1351.05
$even_1$	28.05	1363.06
$even_2$	27.31	1351.99

Table 3.11
Scalability Performance for the *"Soccer"* Sequence.

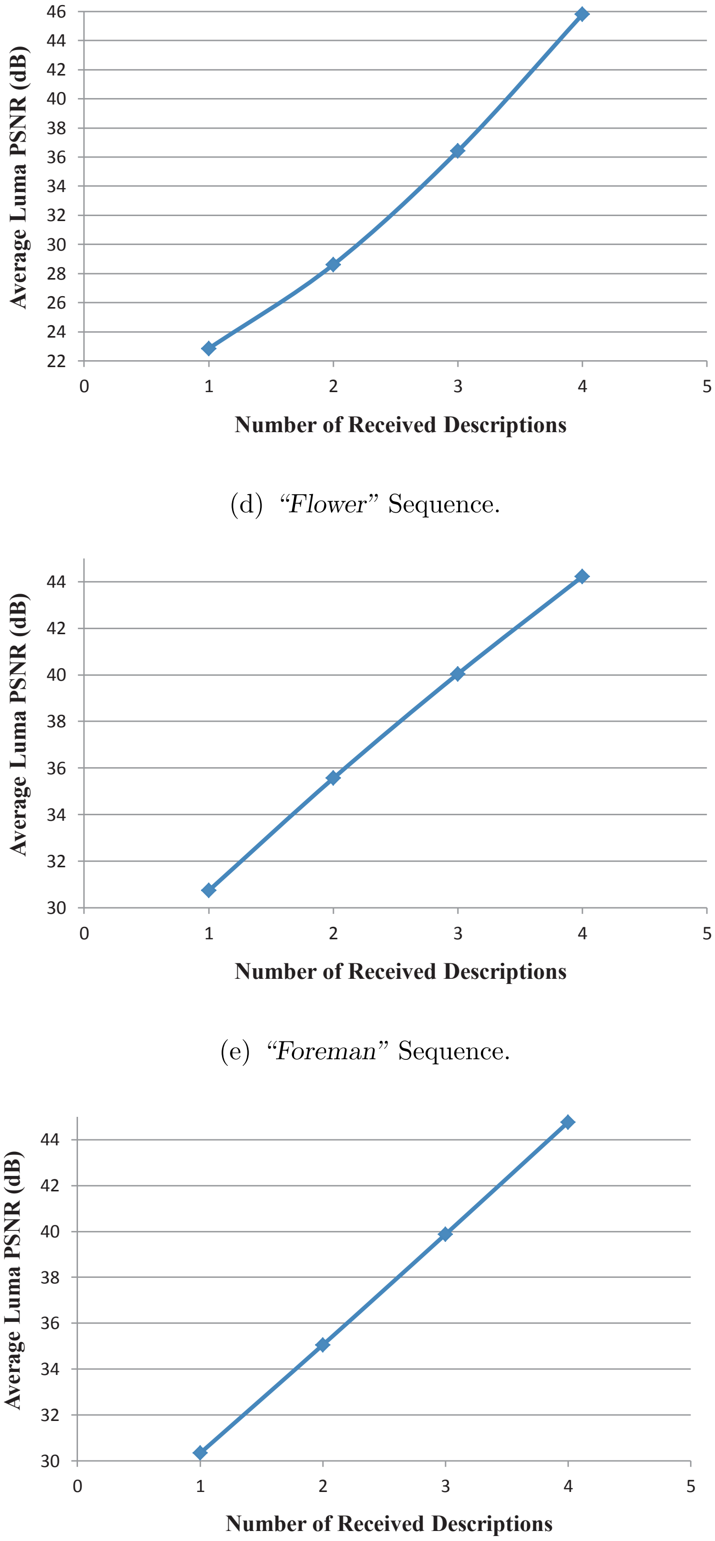
Received Descriptions	PSNR (dB)	Data Rate (kbps)
$odd_1 + odd_2 + even_1 + even_2$	44.36	4564.95
$odd_2 + even_1 + even_2$	40.59	3436.13
$odd_1 + even_1 + even_2$	40.61	3436.23
$odd_1 + odd_2 + even_2$	40.61	3410.40
$odd_1 + odd_2 + even_1$	40.61	3412.09
$odd_1 + odd_2$	31.94	2257.54
$even_1 + even_2$	31.96	2307.41
$odd_1 + even_1$	36.86	2283.37
$odd_2 + even_2$	36.85	2281.58
$odd_1 + even_2$	36.86	2281.68
$odd_2 + even_1$	36.85	2283.27
odd_1	28.27	1128.82
odd_2	28.26	1128.72
$even_1$	28.30	1154.55
$even_2$	28.30	1152.86



(a) *"Bridge"* Sequence.

(b) *"Akiyo"* Sequence.

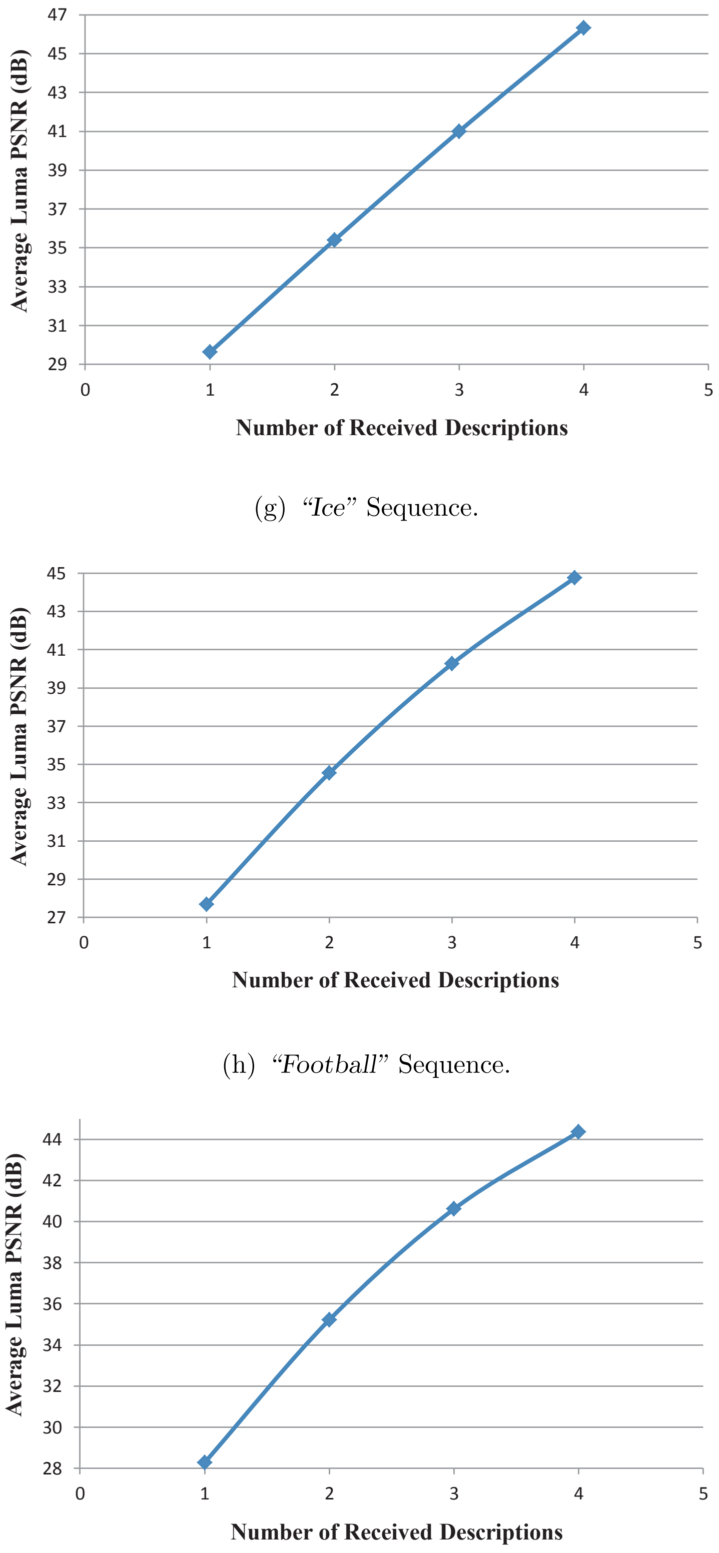
(c) *"Mother"* Sequence.



(d) *"Flower"* Sequence.

(e) *"Foreman"* Sequence.

(f) *"Crew"* Sequence.



(g) *"Ice"* Sequence.

(h) *"Football"* Sequence.

(i) *"Soccer"* Sequence.

Fig. 3.4. Scalability Performance of the Proposed Method.

3.4.2 Packet Loss Performance

In this section, the Gilbert model is used to simulate packet. As discussed in Section 3.3, when P_B is small, L_B is large; and vice versa [99]. The parameters for Gilbert model in our experiments are listed in Table 3.12.

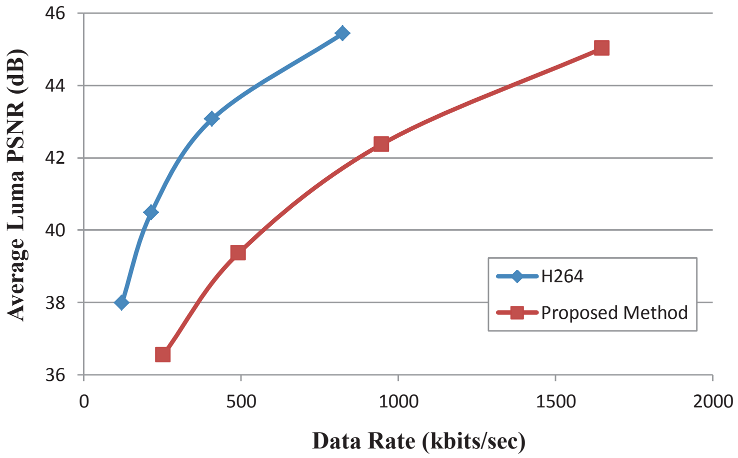
Table 3.12
Gilbert Model Parameters for Various Packet Loss Rates

Loss Rate	5%	10%	15%	20%	25%
Burst Length	5	5	4	4	3

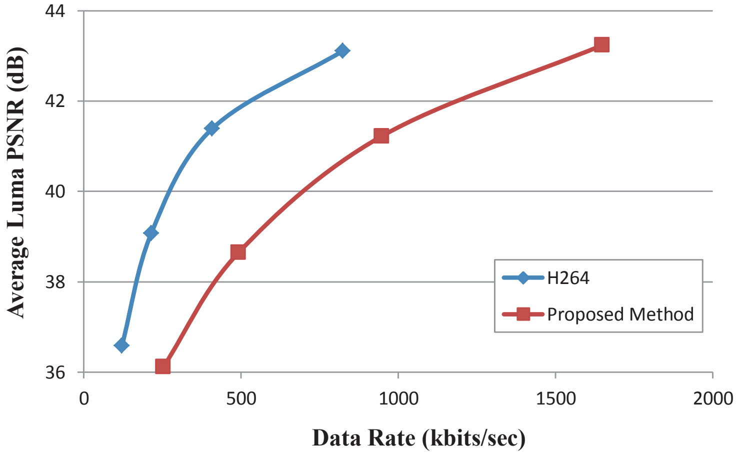
Figure 3.5 to Figure 3.13 show packet loss performance of the 9 testing sequences. Packet loss rates are from 5% to 25%.

As shown, H.264 performs better than the proposed method in terms of coding efficiency. This is because, contrary to motion compression, error resilient video coding usually adds some redundancy in the bitstream to improve the error robustness, which inevitably decreases the coding efficiency. In our case, each description has an independent prediction loop, and the information used as the reference for both intra prediction and inter prediction is reduced. Therefore, the prediction precision degrades and the residue increases which cost more bits to code.

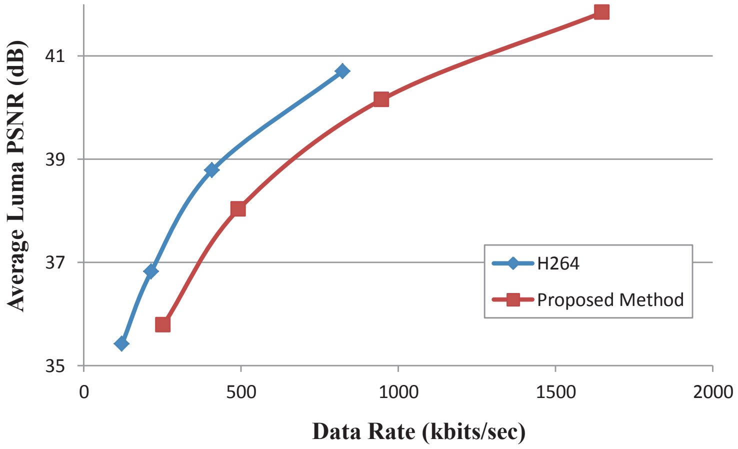
However, when packet loss occurs, H.264 degrades severely compared with our proposed method. Its packet loss performance is obviously worse when the packet loss rate reaches 15% as shown in Figure 3.5 to Figure 3.11. This is because, four descriptions in our MDC model are sent to the decoder through different channels, which are unlikely to have the same lost. For each frame in the video sequence, as long as one description is received, the reconstructed frame at the decoder has an acceptable quality. However for H.264, when a frame is lost, the decoder can only use frame repeat to recover the lost frame. Also note that the improvement for high-motion sequences such as *"Football"* are more oblivious than the improvement for low-to-medium-motion sequences such as *"Mother"*. Similar to the scalability performance, the reason for this is that spatial concealment works better for medium-to-high-motion sequences and temporal concealment works better for low-to-medium-motion sequences. As shown in Figure 3.12 and Figure 3.13, the proposed method does not work well for low-motion sequences. According to Table 3.3 and Table 3.4, temporal concealment works much better than spatial concealment for these two low-motion sequences. Therefore, it is not advantageous to use spatial concealment as the default method for low-motion sequences.



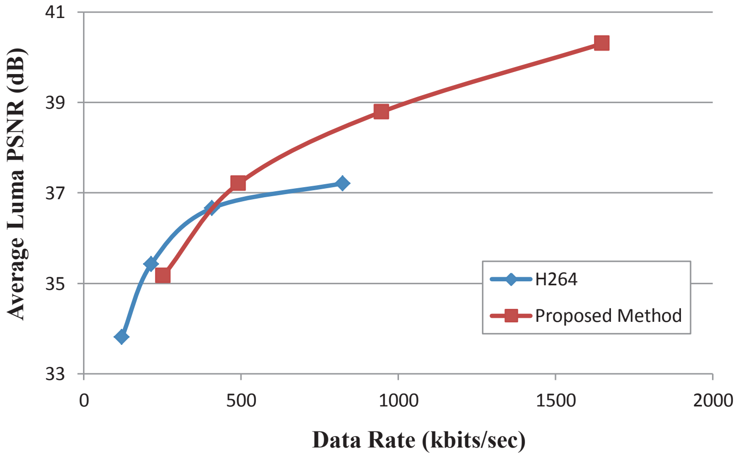
(a) Packet Loss = 0%.



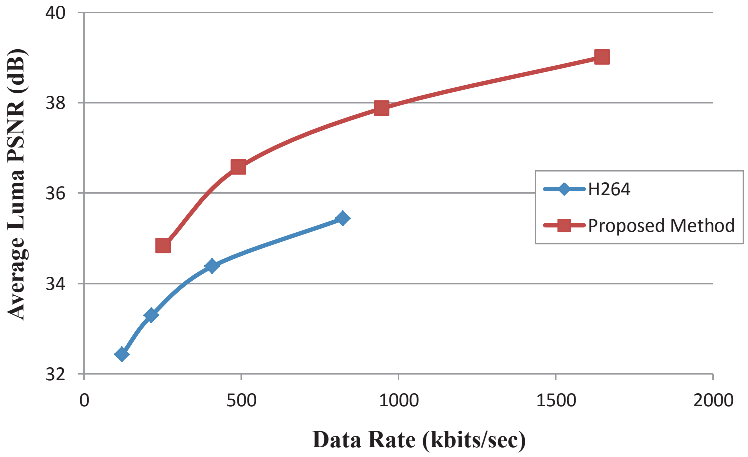
(b) Packet Loss = 5%.



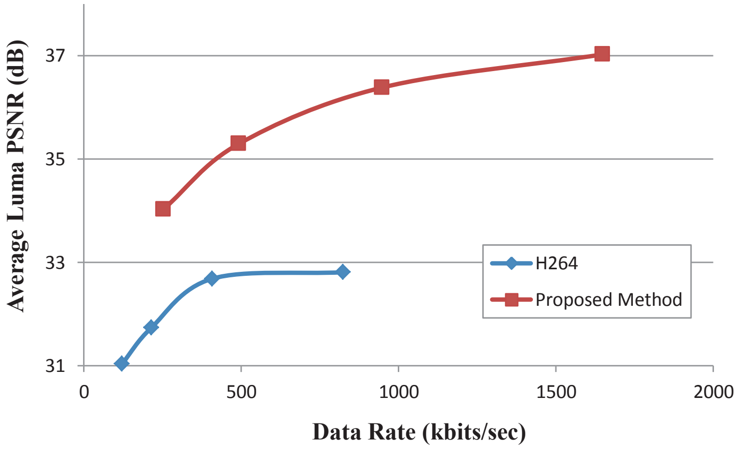
(c) Packet Loss = 10%.



(d) Packet Loss = 15%.

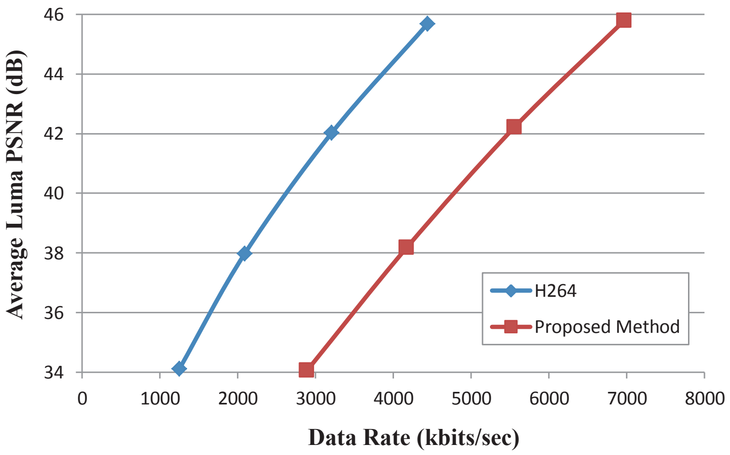


(e) Packet Loss = 20%.

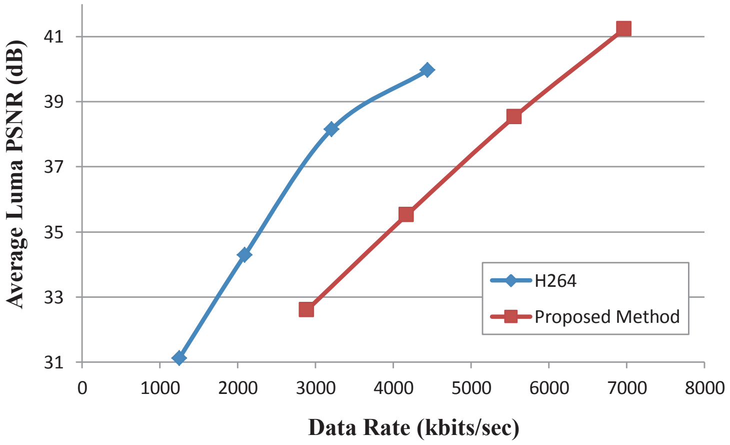


(f) Packet Loss = 25%.

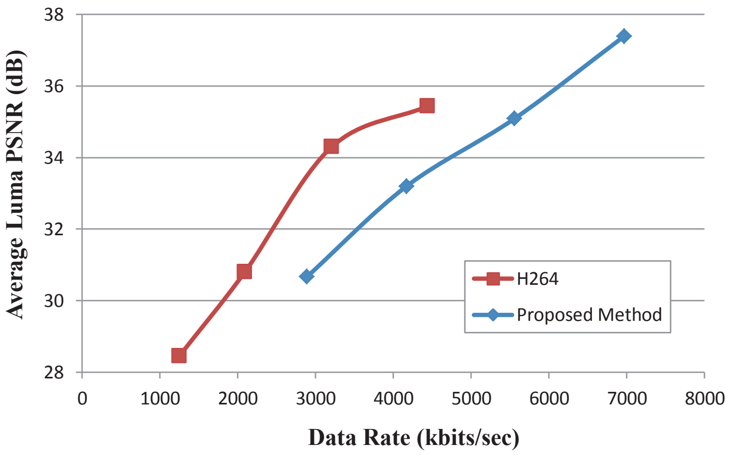
Fig. 3.5. Packet Loss Performances Comparison for the *“Mother”* Sequence.



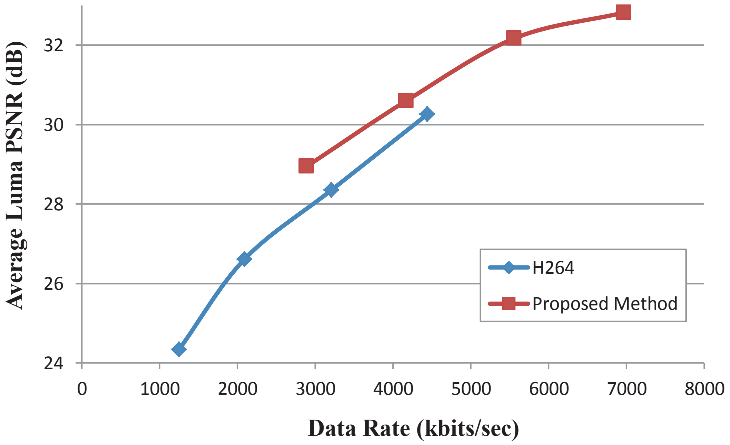
(a) Packet Loss = 0%.



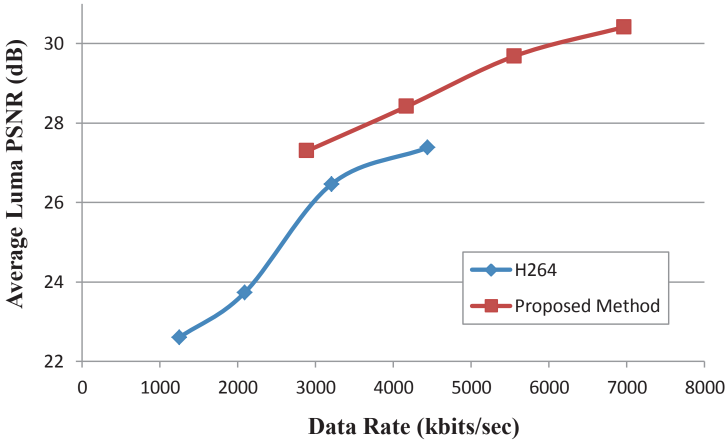
(b) Packet Loss = 5%.



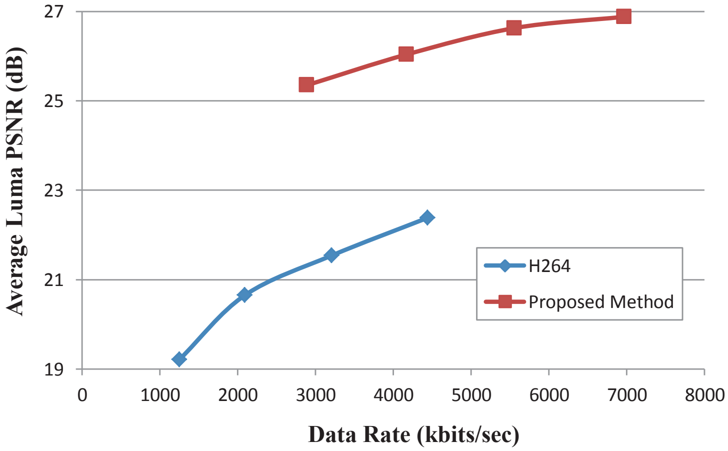
(c) Packet Loss = 10%.



(d) Packet Loss = 15%.

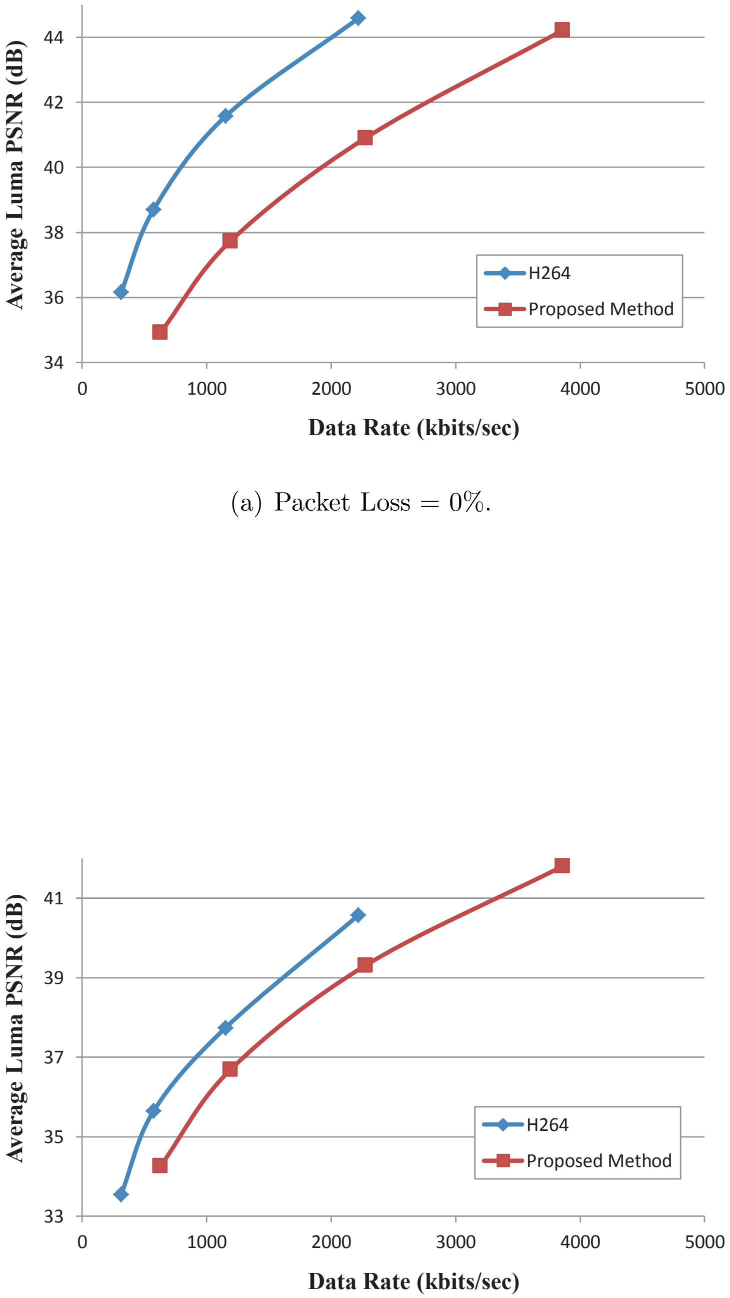


(e) Packet Loss = 20%.



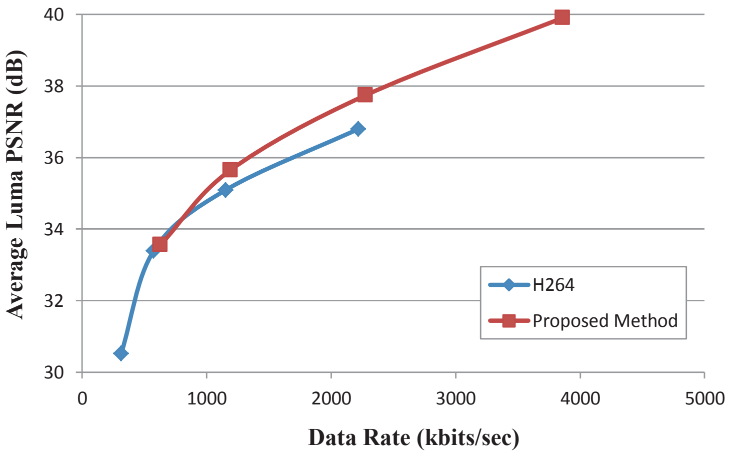
(f) Packet Loss = 25%.

Fig. 3.6. Packet Loss Performances Comparison for the *“Flower”* Sequence.

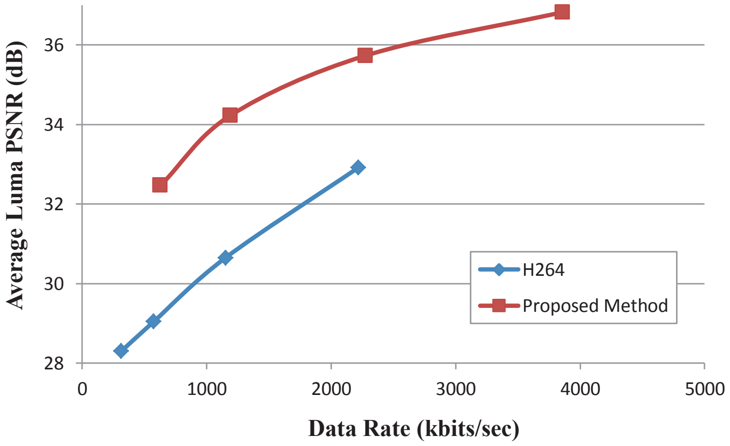


(a) Packet Loss = 0%.

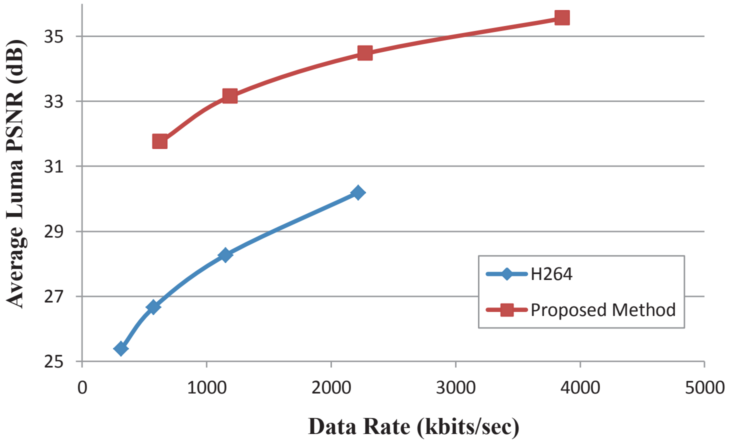
(b) Packet Loss = 5%.



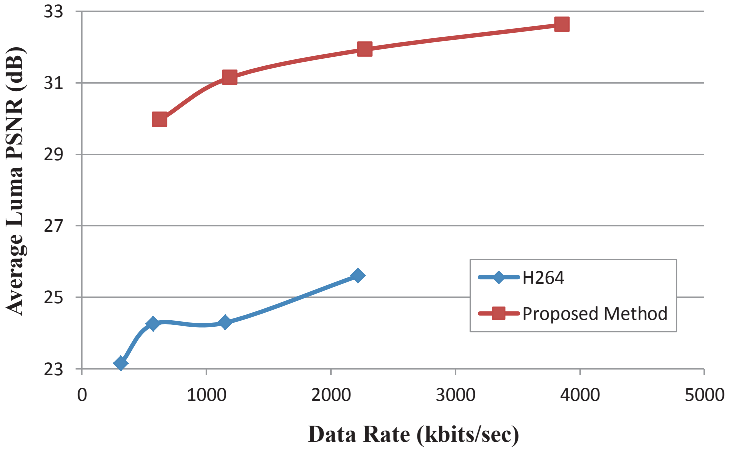
(c) Packet Loss = 10%.



(d) Packet Loss = 15%.

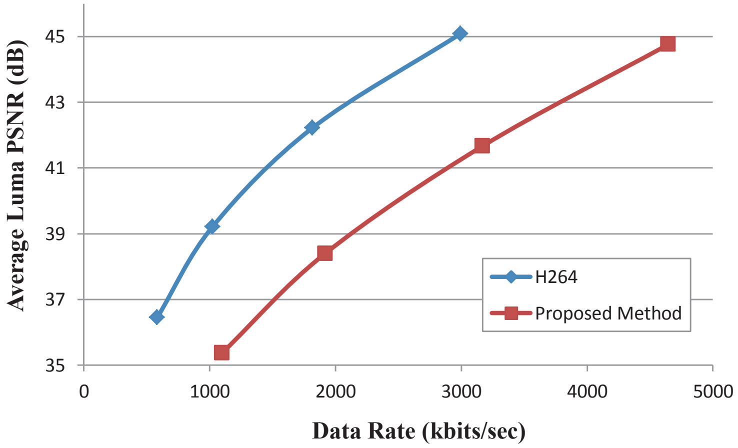


(e) Packet Loss = 20%.

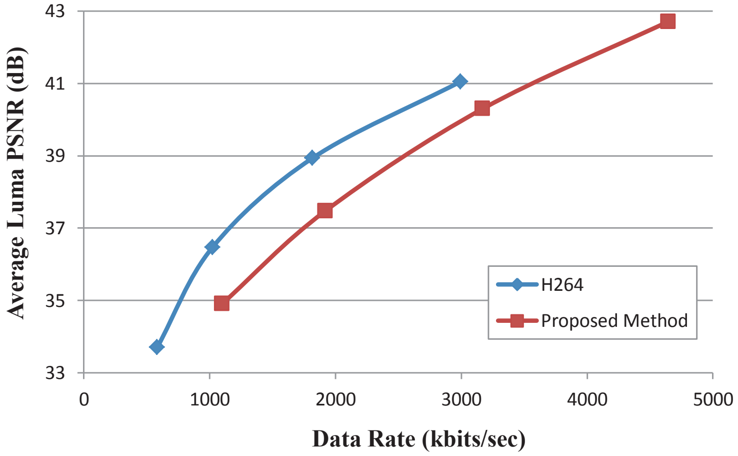


(f) Packet Loss = 25%.

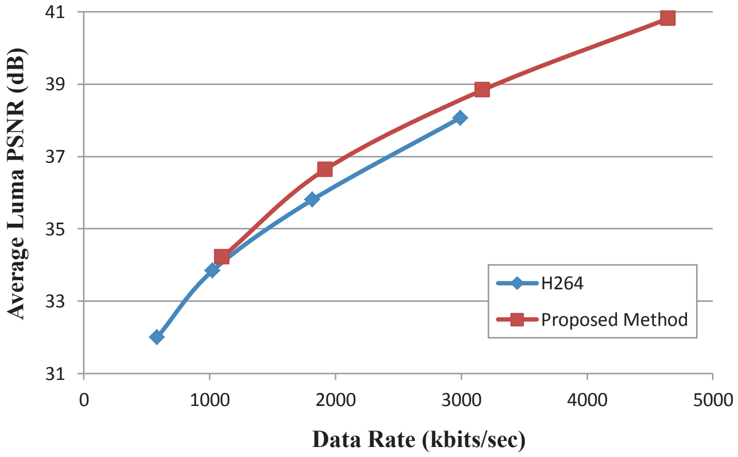
Fig. 3.7. Packet Loss Performances Comparison for the *"Foreman"* Sequence.



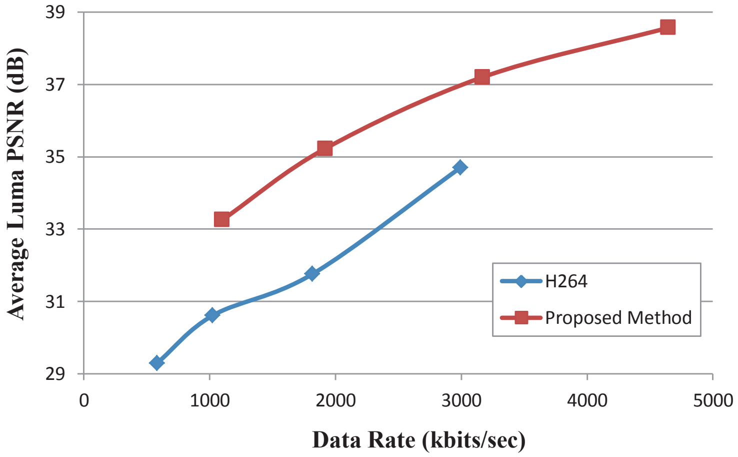
(a) Packet Loss = 0%.



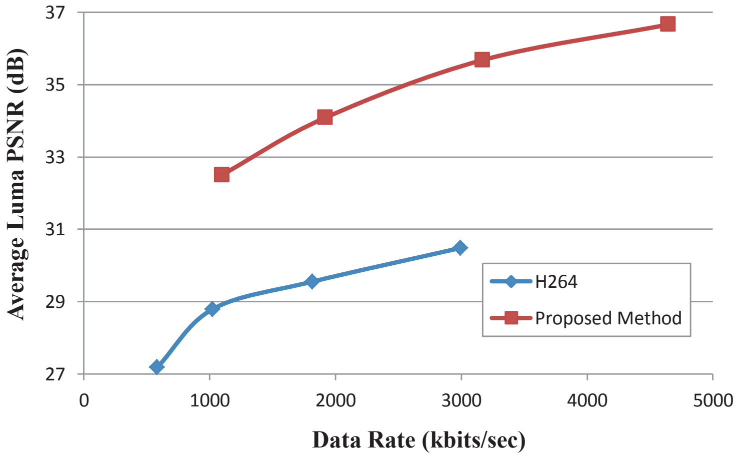
(b) Packet Loss = 5%.



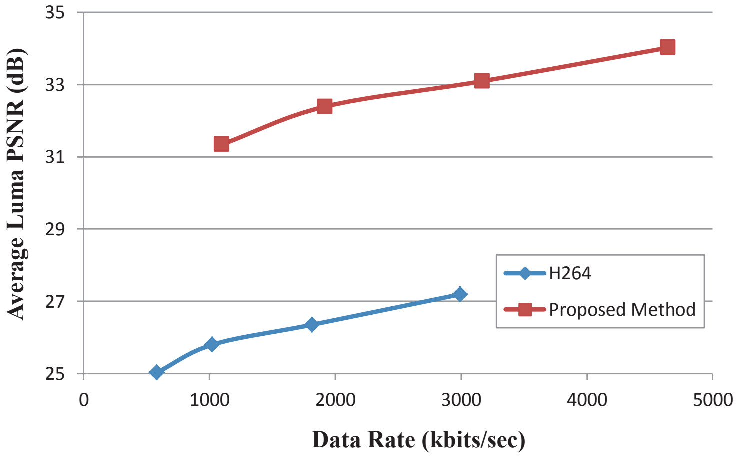
(c) Packet Loss = 10%.



(d) Packet Loss = 15%.

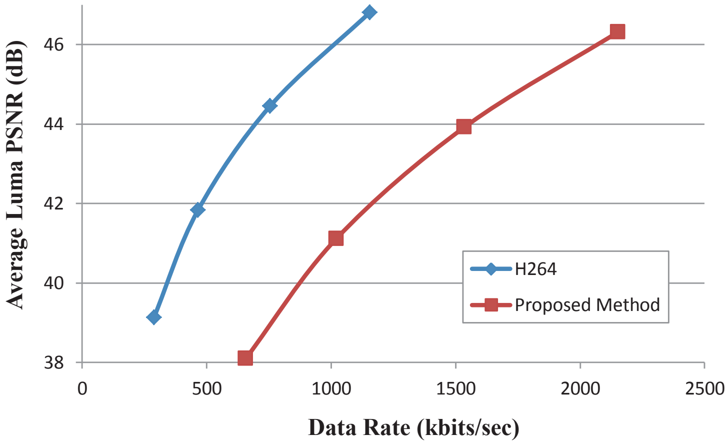


(e) Packet Loss = 20%.

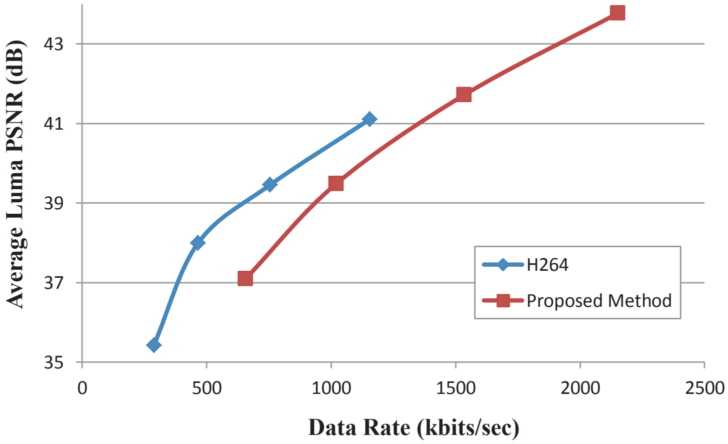


(f) Packet Loss = 25%.

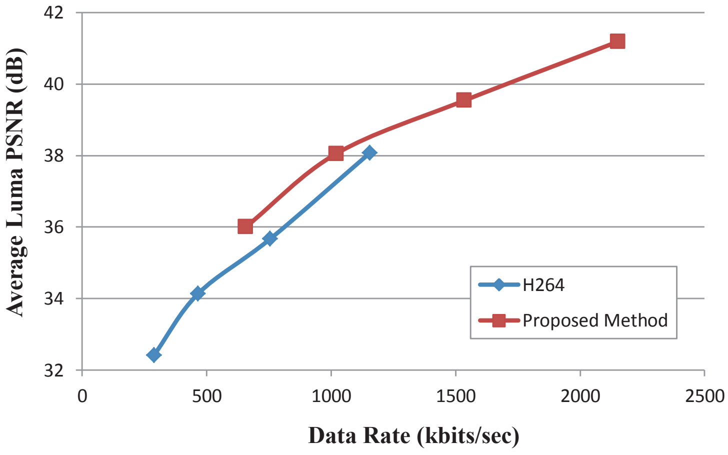
Fig. 3.8. Packet Loss Performances Comparison for the *"Crew"* Sequence.



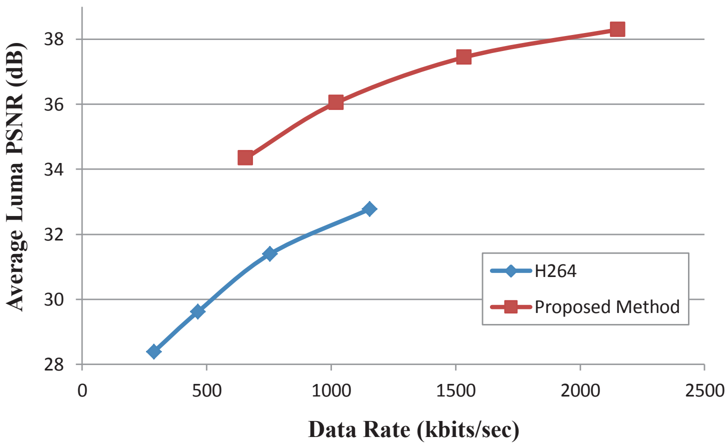
(a) Packet Loss = 0%.



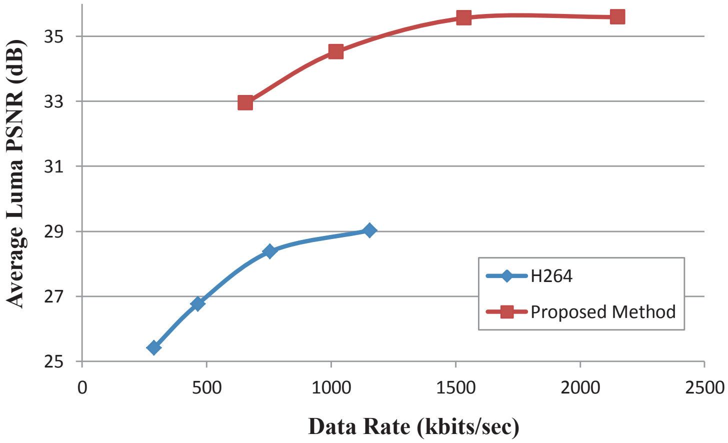
(b) Packet Loss = 5%.



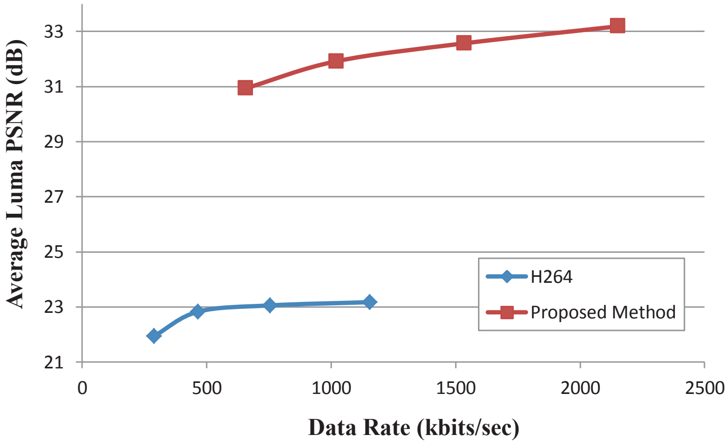
(c) Packet Loss = 10%.



(d) Packet Loss = 15%.

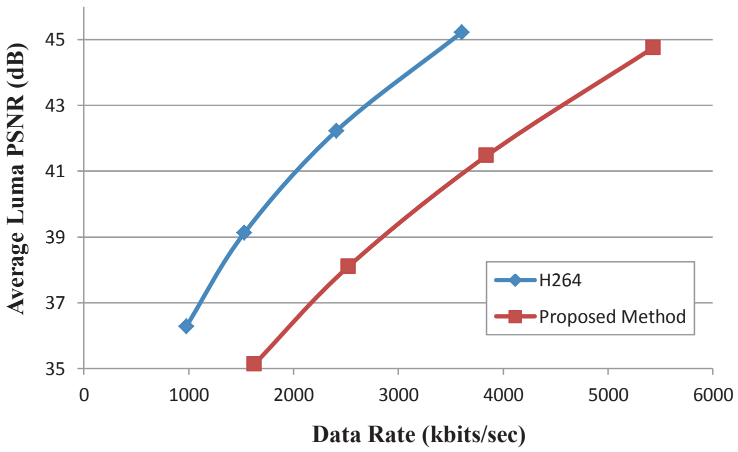


(e) Packet Loss = 20%.

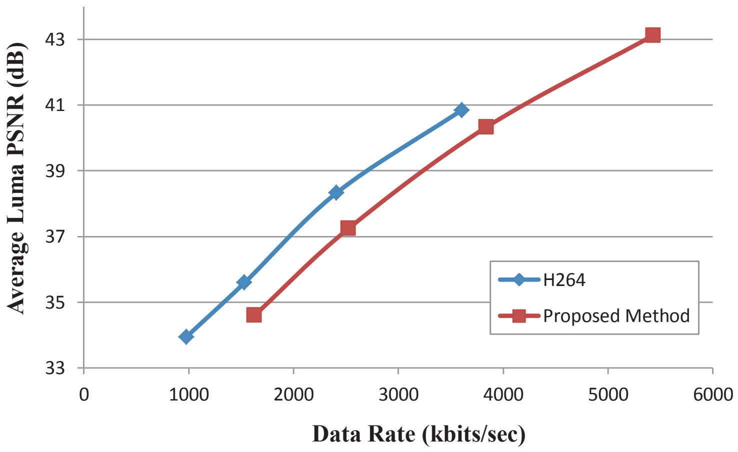


(f) Packet Loss = 25%.

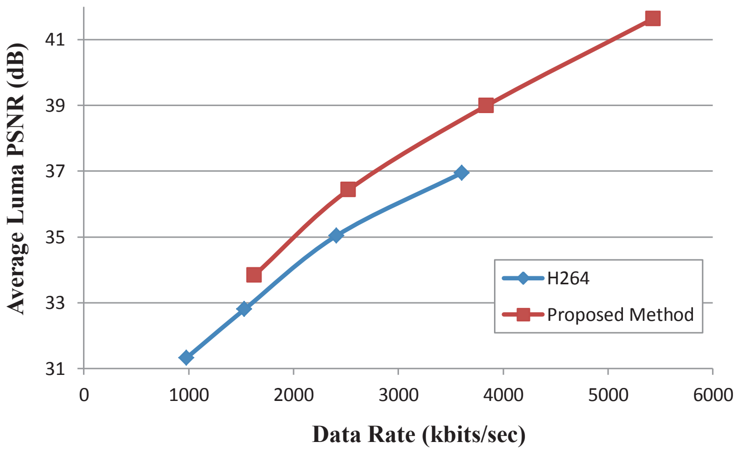
Fig. 3.9. Packet Loss Performances Comparison for the *"Ice"* Sequence.



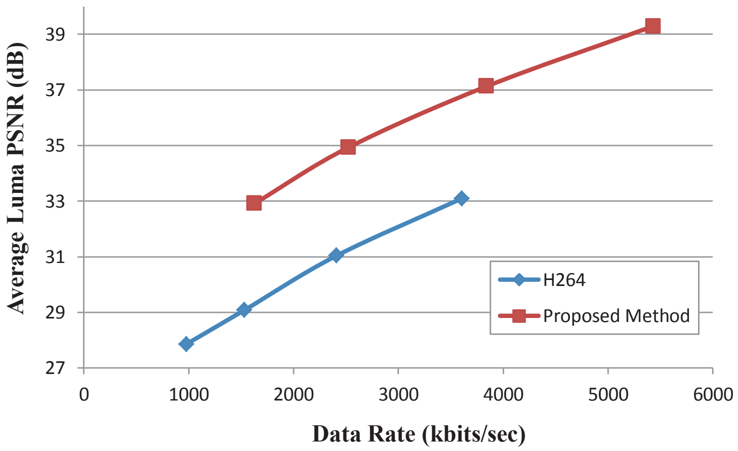
(a) Packet Loss = 0%.



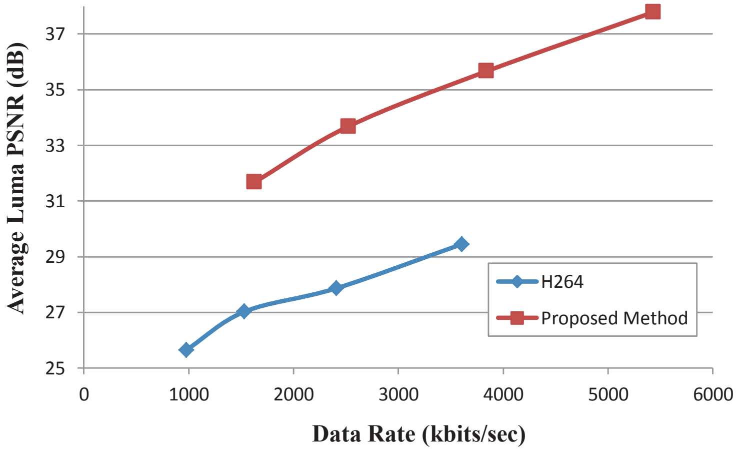
(b) Packet Loss = 5%.



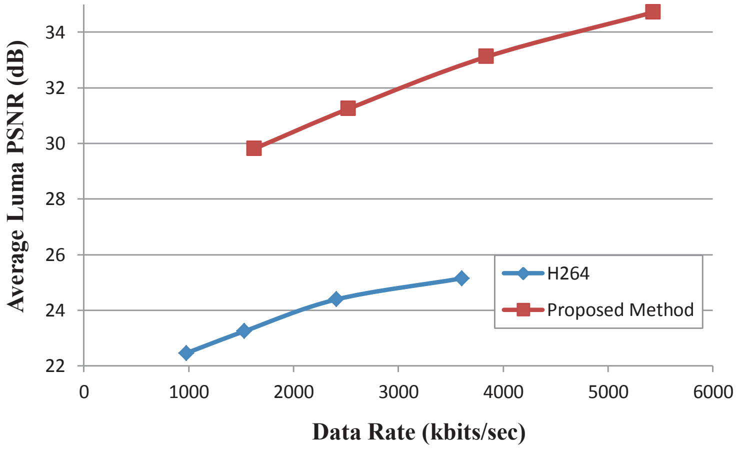
(c) Packet Loss = 10%.



(d) Packet Loss = 15%.

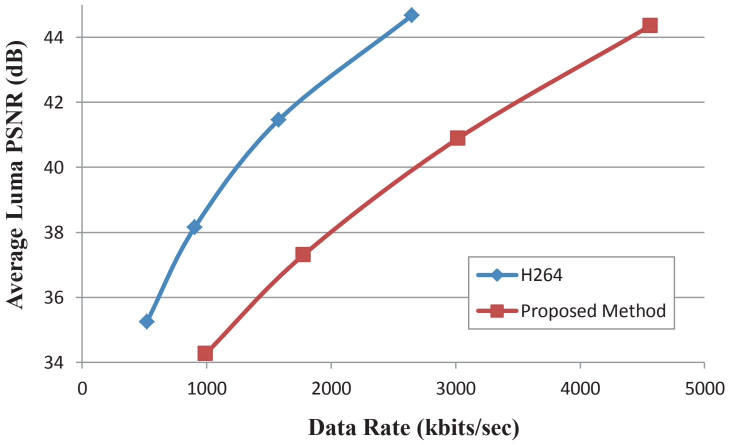


(e) Packet Loss = 20%.

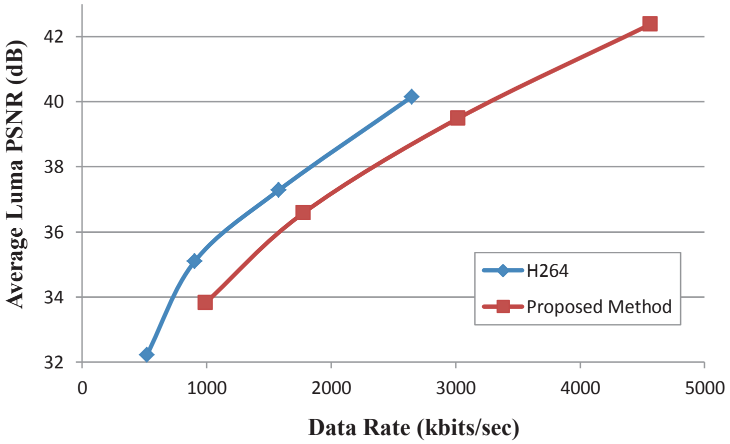


(f) Packet Loss = 25%.

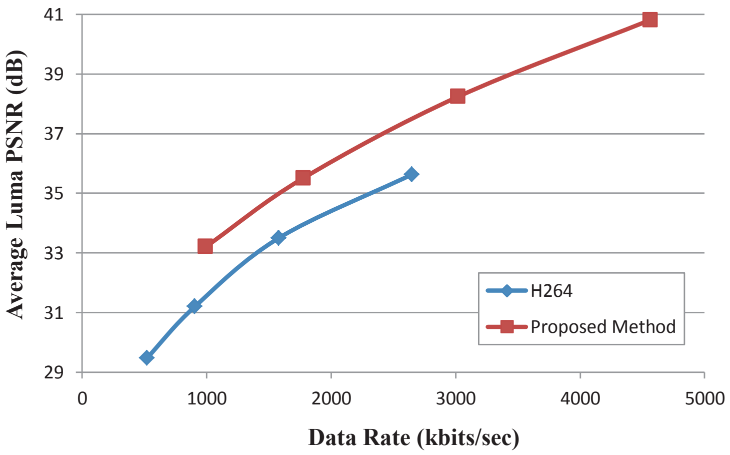
Fig. 3.10. Packet Loss Performances Comparison for the *"Football"* Sequence.



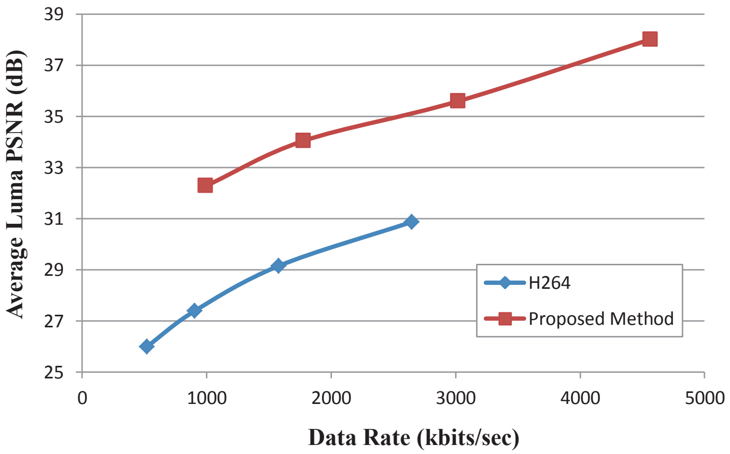
(a) Packet Loss = 0%.



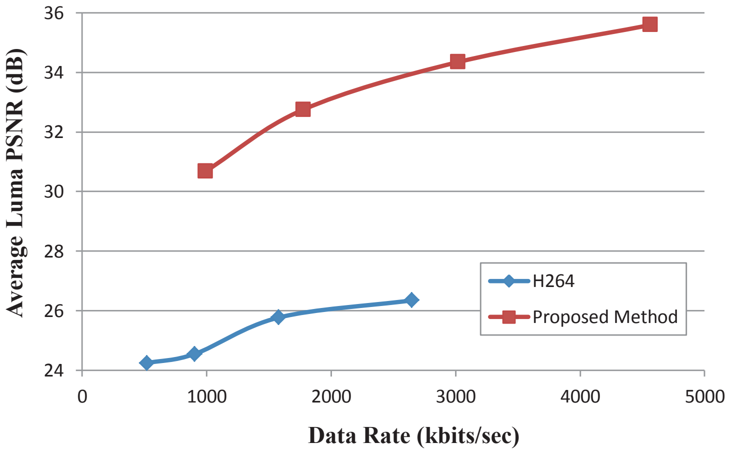
(b) Packet Loss = 5%.



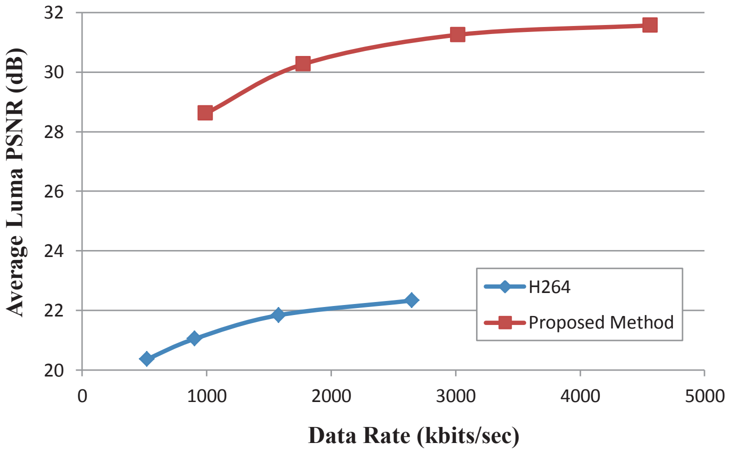
(c) Packet Loss = 10%.



(d) Packet Loss = 15%.

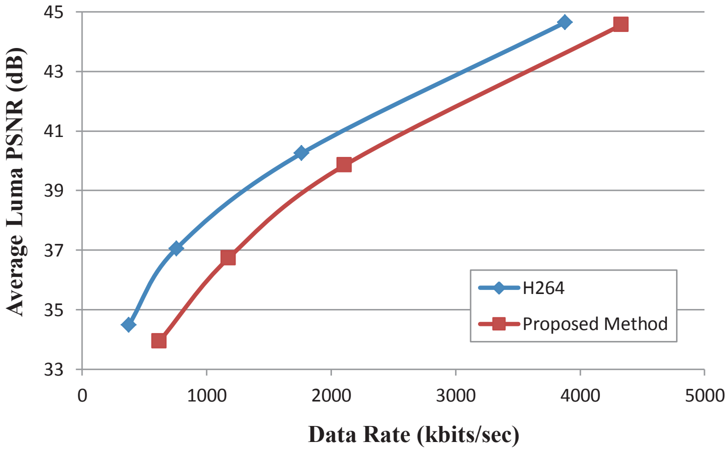


(e) Packet Loss = 20%.

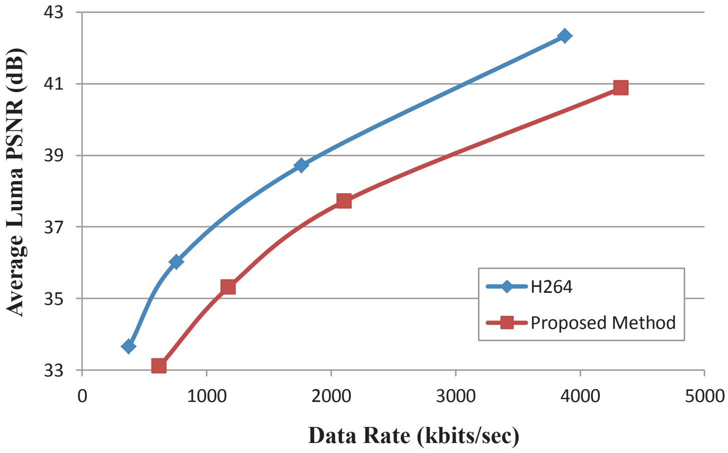


(f) Packet Loss = 25%.

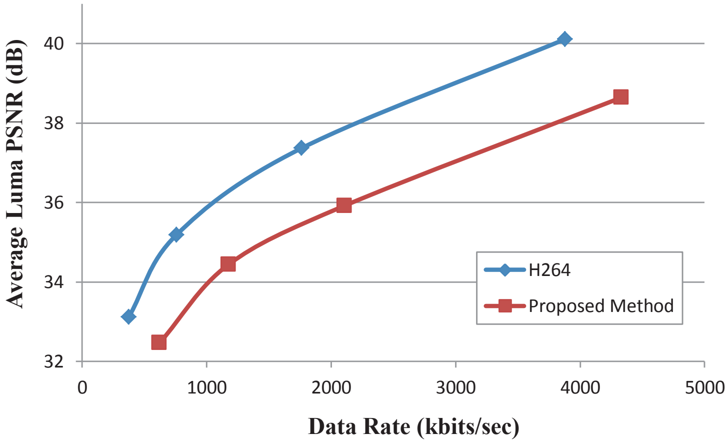
Fig. 3.11. Packet Loss Performances Comparison for the "*Soccer*" Sequence.



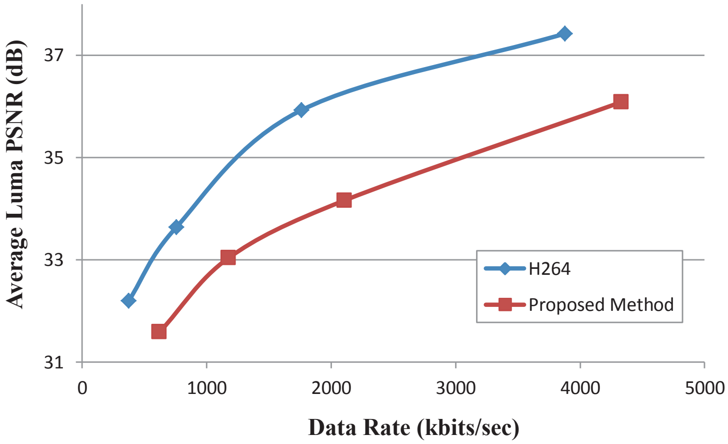
(a) Packet Loss = 0%.



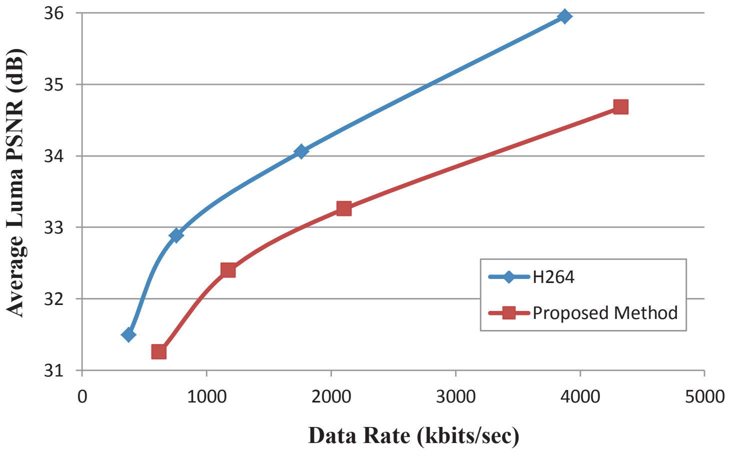
(b) Packet Loss = 5%.



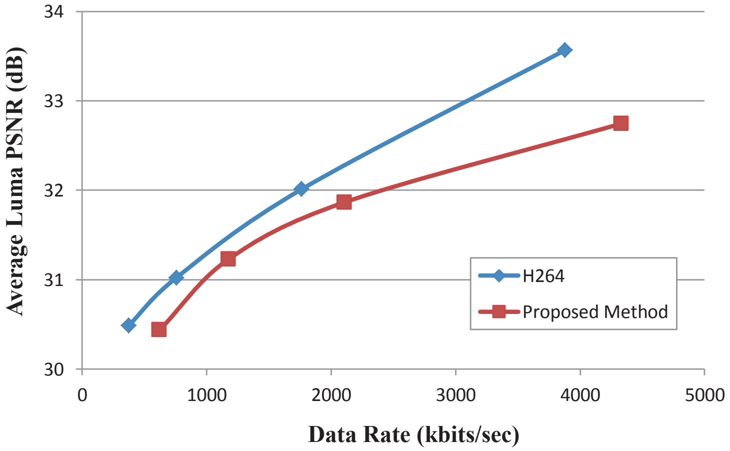
(c) Packet Loss = 10%.



(d) Packet Loss = 15%.

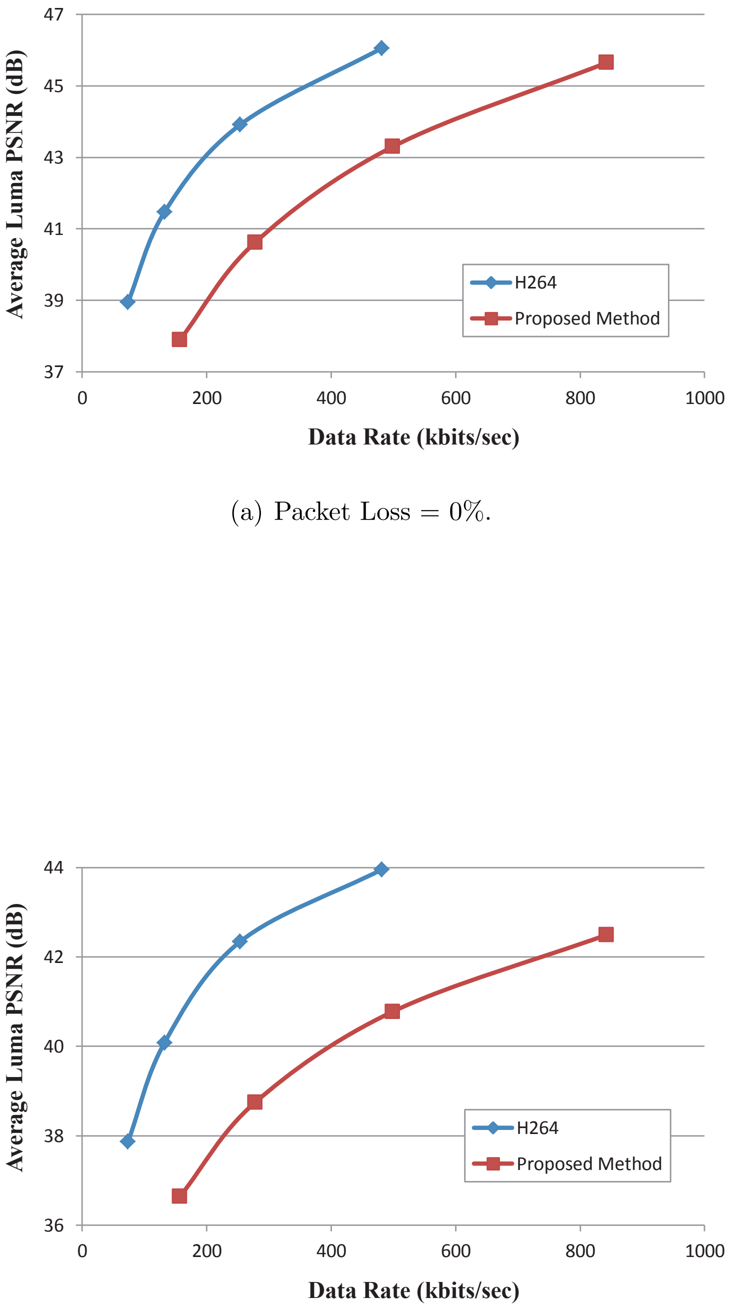


(e) Packet Loss = 20%.



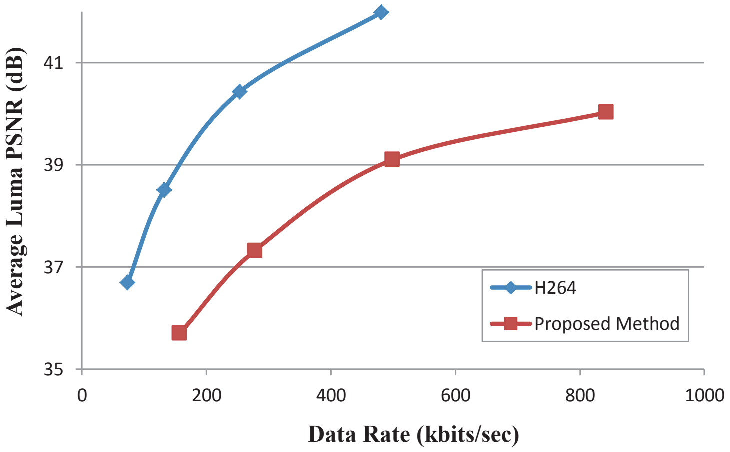
(f) Packet Loss = 25%.

Fig. 3.12. Packet Loss Performances Comparison for the *"Bridge"* Sequence.

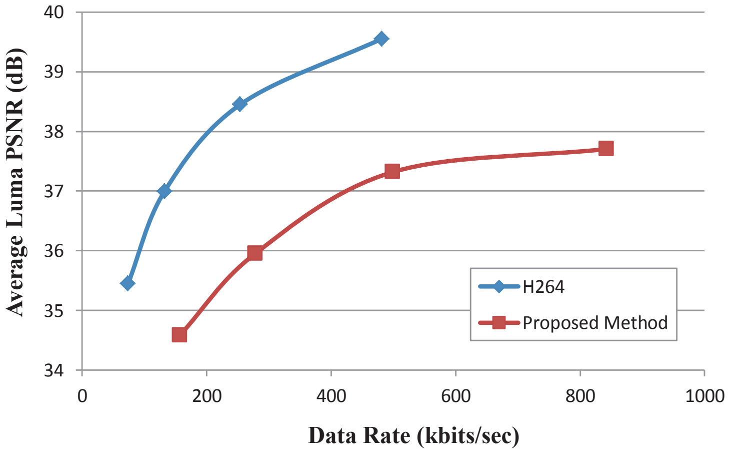


(a) Packet Loss = 0%.

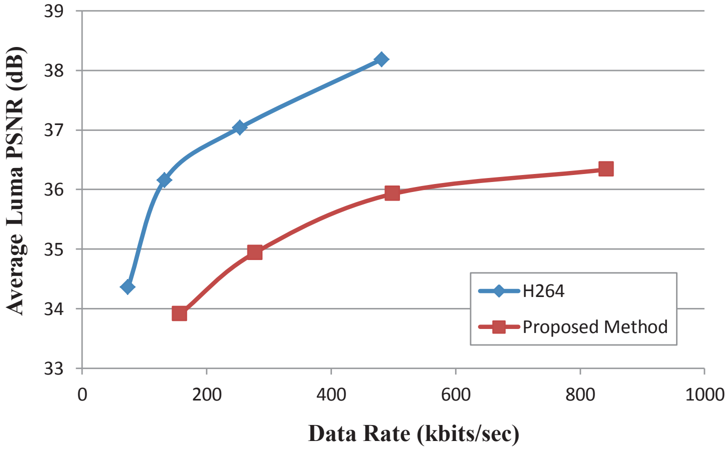
(b) Packet Loss = 5%.



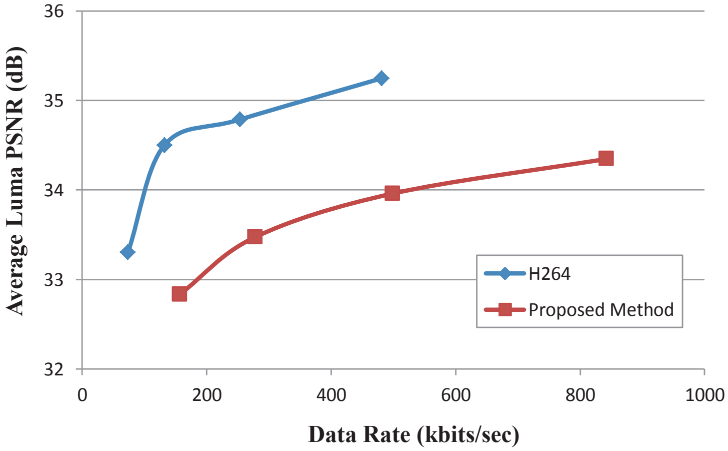
(c) Packet Loss = 10%.



(d) Packet Loss = 15%.



(e) Packet Loss = 20%.



(f) Packet Loss = 25%.

Fig. 3.13. Packet Loss Performances Comparison for the *"Akiyo"* Sequence.

3.5 Conclusion

In this chapter, we proposed a MDC method with four descriptions based on both temporal and spatial correlations. Experimental results demonstrate the efficacy of the proposed method in two aspects. For scalability performance, the proposed method provides a graceful degradation with a decrease in the number of received descriptions. For packet loss performance, the proposed method outperforms H.264 in terms of error robustness.

In this work, the error concealment scheme is fixed as illustrated in Section 3.2, i.e., spatial concealment is the default method and temporal concealment is the secondary method. This scheme works well for medium-to-high-motion sequences, such as the *"Football"* and *"Soccer"* sequences, but does not work well for low–motion sequences such as the *"Bridge"* and *"Akiyo"* sequence. An extension work is to develop an adaptive error concealment scheme that chooses between temporal concealment and spatial concealment according to the characteristics of video sequences which is presented in the next chapter.

4. AN ADAPTIVE ERROR CONCEALMENT FOR MDC BASED ON ERROR TRACKING

The work described in Chapter 3 uses spatial concealment as the default error concealment method, and temporal concealment as the secondary method. This scheme works well for high-motion sequences, but is weak in terms of error robustness for low-motion sequences. This is because in some circumstances, temporal concealment can be more effective than spatial concealment. In this chapter we propose an adaptive spatial-temporal error concealment method to address with this problem. We use the same MDC partition architecture as in Chapter 3 and take into account the distortion due to both error concealment and error propagation to choose the best method between spatial concealment and temporal concealment. This chapter is organized as follows: Section 4.1 provides a detailed description of the proposed method [100]. Section 4.2 describes the error concealment schemes for the proposed method. Experimental results and discussions are provided in Section 4.3. And we conclude this chapter in Section 4.4.

4.1 Distortion Analysis

At the decoder, the distortion of a frame consists of two parts: distortion from initial concealment (i.e., error due to the current loss), and distortion from the frame used for error concealment (i.e., propagation from previous errors). Note that one sequence is made up of four representations. The term “frame” in the following subsections refers to a subframe in one of the four representations and each subframe is encoded into one packet.

4.1.1 Initial Concealment Distortion

The initial concealment distortion of frame k, denoted as D_k^c, is the distortion that would result between the original frame k and the concealed frame k when the reference frame for concealment is error free. Here we use squared error to represent the distortion. For a sequence with the resolution of $m \times n$, the initial concealment distortion for frame k is:

$$D_k^c = \frac{1}{mn} \times \sum_{y=0}^{m-1} \sum_{x=0}^{n-1} \left(\hat{f}_k(x,y) - f'_k(x,y)\right)^2 \tag{4.1}$$

where $\hat{f}_k(x,y)$ is the reconstructed pixel of frame k at the encoder, and $f'_k(x,y)$ is the concealed pixel of frame k when the frame used for concealment is error free. For each MDC representation D_k^c is obtained at the encoder for each of temporal concealment and spatial concealment for every packet. This information is transmitted along with the encoded bitstream to help the decoder choose the optimal concealment method. The decision of spatial or temporal concealment is made at the frame level.

4.1.2 Distortion When a Frame is Received

To measure the distortion of a frame received at the decoder, we have to analyze the effect of error propagation. For simplicity, we assume the GOP structure is IPPP... with single frame reference. Assume pixel (x,y) of P-frame k is predicted from pixel $(\tilde{x},\tilde{y})$ of frame $k-1$. Note that we assume each of the four representations of a frame is either received in its entirety or not received. We also assume that the four representations are not lost at the same time.

1) For a received I frame, as all the predictions are within the frame, there is no error propagation. Since the decoder can decode the frame exactly, the pixel error for frame k is:

$$e_k(x,y) = 0 \tag{4.2}$$

2) For P frame, we assume the reconstructed pixel of frame k at the decoder is $\tilde{f}_k(x,y)$ and the received corresponding residue is $\hat{d}_k(x,y)$, then we have:

$$\tilde{f}_k(x,y) = \tilde{f}_{k-1}(\tilde{x},\tilde{y}) + \hat{d}_k(x,y) \tag{4.3}$$

Therefore the error is:

$$\begin{aligned} e_k(x,y) &= \hat{f}_k(x,y) - \tilde{f}_k(x,y) \\ &= (\hat{f}_{k-1}(\tilde{x},\tilde{y}) + \hat{d}_k(x,y)) - (\tilde{f}_{k-1}(\tilde{x},\tilde{y}) + \hat{d}_k(x,y)) \\ &= \hat{f}_{k-1}(\tilde{x},\tilde{y}) - \tilde{f}_{k-1}(\tilde{x},\tilde{y}) \\ &= e_{k-1}(\tilde{x},\tilde{y}) \end{aligned} \tag{4.4}$$

In other words, when a P frame is received, its pixel error is the propagated error from its reference frame.

4.1.3 Distortion When a Frame is Lost

If a frame is lost at the decoder, error concealment is used. We assume that $g(x,y)$ is the pixel value at (x,y) from the frame used for concealment, $\hat{g}(x,y)$ is its reconstructed form at the encoder and $\tilde{g}(x,y)$ is its reconstructed form at the decoder. Note that $g(x,y)$ refers to a frame from one of the other MDC descriptions.

1) For temporal concealment, we use a non-motion compensated method which copies the pixel value of the same position from the frame used for concealment. Thus, the concealed pixel of frame k, $\tilde{f}'_k(x,y)$ equals to $\tilde{g}_k(x,y)$. Therefore,

$$\begin{aligned} e_k(x,y) &= \hat{f}_k(x,y) - \tilde{f}'_k(x,y) \\ &= \hat{f}_k(x,y) - \tilde{g}_k(x,y) \\ &= (\hat{f}_k(x,y) - \hat{g}_k(x,y)) + (\hat{g}_k(x,y) - \tilde{g}_k(x,y)) \\ &= e^c_k(x,y) + e^g_k(x,y) \end{aligned} \tag{4.5}$$

Here $e^c_k(x,y)$ denotes the pixel error from initial concealment, and $e^g_k(x,y)$ denotes the pixel error of the frame used for concealment.

2) For spatial concealment, we use a two-neighbor bilinear filter. Thus, $\tilde{f}'_k(x,y) = (\tilde{g}_k(x-1,y) + \tilde{g}_k(x+1,y))/2$. Therefore,

$$\begin{aligned} e_k(x,y) &= \hat{f}_k(x,y) - \tilde{f}'_k(x,y) \\ &= \hat{f}_k(x,y) - (\tilde{g}_k(x-1,y) + \tilde{g}_k(x+1,y))/2 \\ &= (\hat{f}_k(x,y) - (\hat{g}_k(x-1,y) + \hat{g}_k(x+1,y))/2) \\ &+ ((\hat{g}_k(x-1,y) + \hat{g}_k(x+1,y))/2 \\ &- (\tilde{g}_k(x-1,y) + \tilde{g}_k(x+1,y))/2) \\ &= e^c_k(x,y) + (e^g_k(x-1,y) + e^g_k(x+1,y))/2 \end{aligned} \tag{4.6}$$

Summing over all pixels in the frame:

$$D_k = D^c_k + D^g_k \tag{4.7}$$

where D_k is the distortion of frame k to be decoded or concealed, D^c_k is its initial error concealment distortion, and D^g_k is the distortion due to the frame used for concealment.

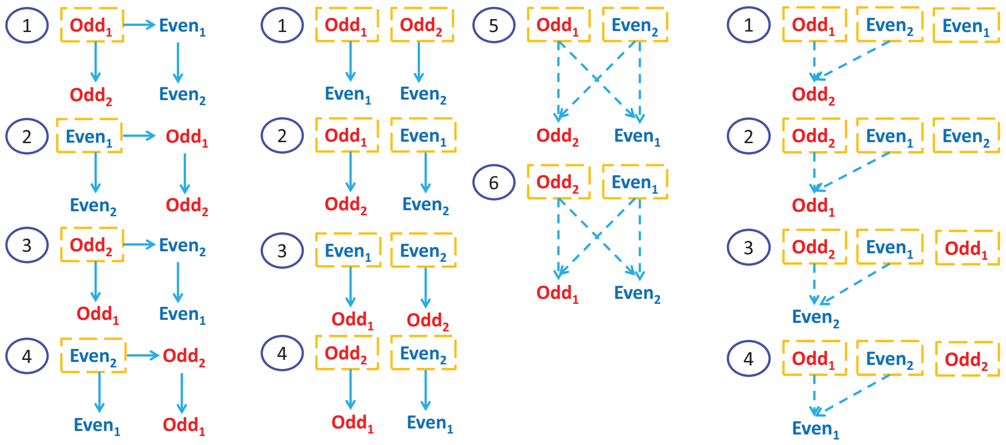
4.2 Adaptable Spatial-Temporal Error Concealment Schemes

4.2.1 Error Concealment Schemes

Figure 4.1 shows the error concealment schemes for each receiving scenario. The yellow frames are the received descriptions. The arrows indicate the relationships between the frames used for concealment and the lost frames. The solid arrows represent fixed concealment decisions, and dashed arrows represent adaptable concealment decisions. For each receiving scenario:

1) When one description is received, we use spatial concealment first and then use temporal correlation to conceal the other two descriptions.

2) When two descriptions are received, three cases are considered. If the two descriptions are from the same spatial correlation such as Odd$_1$ and Odd$_2$, we use temporal concealment to conceal Even$_1$ and Even$_2$. If the two descriptions are from the same temporal correlation such as Odd$_1$ and Even$_1$, we use spatial concealment to conceal



(a) one description received.

(b) two descriptions received.

(c) three descriptions received.

Fig. 4.1. Our Proposed Error Concealment Schemes.

Odd$_2$ and Even$_2$. If the two descriptions are from different temporal and spatial correlations such as Odd$_1$ and Even$_2$, we use the adaptable error concealment method based on error tracking. The concrete procedure is illustrated in the next subsection.
3) When three descriptions are received, we use the adaptable error concealment method based on error tracking to conceal the lost description.

4.2.2 Joint Encoder and Decoder Error Tracking

The procedure for the adaptable error concealment based on error tracking is as follows:

1) Each encoded packet is assumed to be lost at the encoder. The temporal concealment and the spatial concealment are used separately and each corresponding initial error concealment distortion D_k^c is recorded. This information is transmitted along with the encoded bitstream to the decoder. Table 4.1 illustrates how many bits are used to represent the distortion information for a low-motion sequence *"Bridge"*, a medium-motion sequence *"Mother"* and a high-motion sequence *"Football"*. Com-

pared to the original data rate, the additional data rate caused by the overhead can be ignored.

2) After each received frame is decoded, its distortion D_k is updated according to Equation 4.2 to 4.6.

3) When packet loss occurs, the distortion of choosing each concealment method is the summation of its corresponding initial concealment distortion and the distortion of the frame used for concealment as illustrated in Equation 4.7.

4) The concealment method with the smaller distortion from step 3) is chosen, and the corresponding distortion D_k is updated.

Table 4.1
Additional data caused by initial error concealment distortion information

sequence	*Bridge*	*Mother-Daughter*	*Football*
bits to code a distortion	7	7	11
additional data rate (kbps)	0.84	0.84	1.32
original data rate (kbps)	185.86-1938.72	108.35-760.19	833.21-3540.34

4.3 Experimental Results

In this section, we evaluate the packet loss performance of the proposed method. The results are compared with our non-adaptive method described in Chapter 3.

All the experiments are implemented by modifying the JVT JM software version 16.1. The testing sequences have original resolutions of CIF (352 × 288) at 30 frames/sec with 200-frame length. Thus each description has 100 subframes at 15 frames/sec. The coding structure is "IPPP...", with an I-frame refreshment every

15 subframes in each description. The quantization parameters for I frames and P frames are 24, 28, 32 and 36.

Nine testing sequences are used in our experiments: *"Bridge"*, *"Akiyo"*, *"Container"*, *"Mother"*, *"Paris"*, *"News"*, *"Ice"*, *"Soccer"* and *"Football"*. Among these testing sequences, *"Bridge"*, *"Akiyo"* and *"Container"* are low-motion sequences. *"Mother"*, *"Paris"* and *"News"* are medium-motion sequences. *"Ice"*, *"Soccer"* and *"Football"* are high-motion sequence. Experimental results are obtained by averaging 100 channel transmission simulations using the Gilbert model. As noted in Section 3.3, when P_B is small, L_B is large; and vice versa [99]. The parameters for the Gilbert model in our experiments are shown in Table 4.2.

Table 4.2
Gilbert Model Parameters for Various Packet Loss Rates

Loss Rate	5%	10%	15%	20%	25%	30%
Burst Length	5	5	4	4	3	3

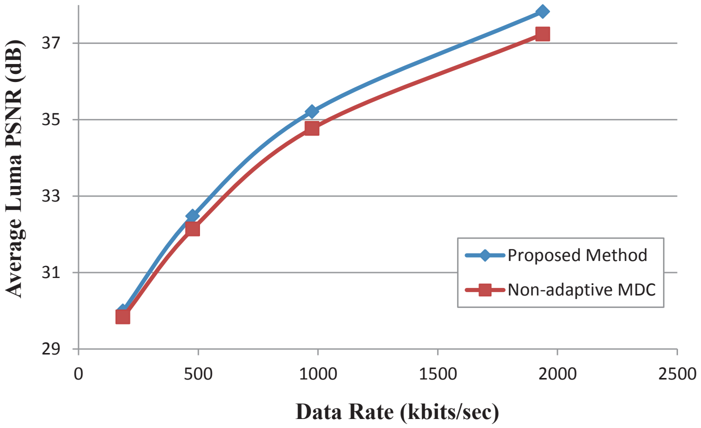
Figure 4.2 to 4.10 show packet loss performances of the testing sequences. Packet loss rates are from 5% to 30%.

Figure 4.2 to Figure 4.4 show that our proposed method has an obvious improvement over the non-adaptive method [95]. The improvement is more obvious with the increase of packet loss rate. When packet loss rate reaches 30%, our proposed method has a gain of up to 2.5 dB. This is because for low-motion sequences, temporal concealment has smaller error concealment distortion and is often chosen as the concealment method in our adaptive method. However for the non-adaptive method, spatial concealment is used as the default method and temporal concealment is the secondary method. Therefore, for low motion sequences such as the *"Bridge"* sequence, our proposed method is much sharper than the non-adaptive method and the overall visual quality is much better.

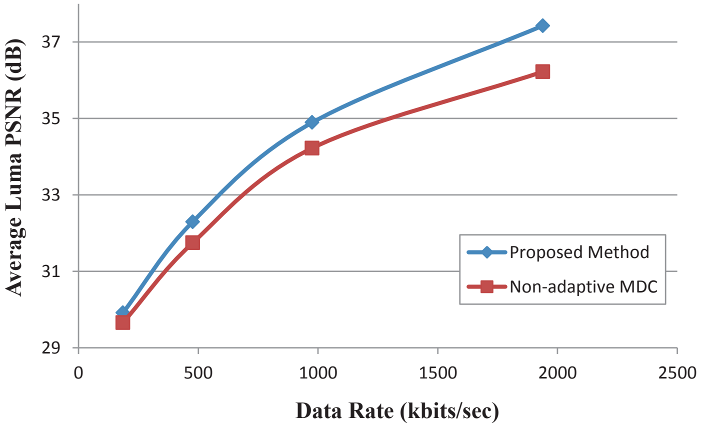
Figure 4.5 to Figure 4.7 show the results for the medium-motion sequences. Figure 4.5 illustrates that the proposed method has an improvement over the non-adaptive method, but not as much as for low-motion sequences. This is because for medium-motion sequences, the difference between the distortions caused by temporal concealment and spatial concealment is not significant. As the error concealment decision is made on distortion due to both error and error propagation, the improvement by using our method is not significant. Figure 4.6 and Figure 4.7 show an obvious improvement over the non-adaptive method, although they are medium-motion sequences. This is because for these two sequences, there are many details in the background. So the difference between the distortions caused by temporal concealment and spatial concealment are significant, and using the correct error concealment method can greatly improve the performance.

Figure 4.8 to Figure 4.10 show the results for high-motion sequences. As shown in each sequence, there is not much improvement here. This is because for high-motion sequences, spatial concealment has smaller error concealment distortion and is often chosen as the concealment method in our adaptive method, which is the same as the non-adaptive method that uses spatial concealment as the default method. When the packet loss rate is 30%, our proposed method is slightly worse than the non-adaptive method. There are two reasons for these inconsistent results. Our proposed method chooses the error concealment method based on the estimated distortion which is less precise when the packet loss rate is high. However, the random characteristic of the transmission channel inevitably influences the experimental results, even though the results are obtained by averaging 100 channel transmission simulations using Gilbert model.

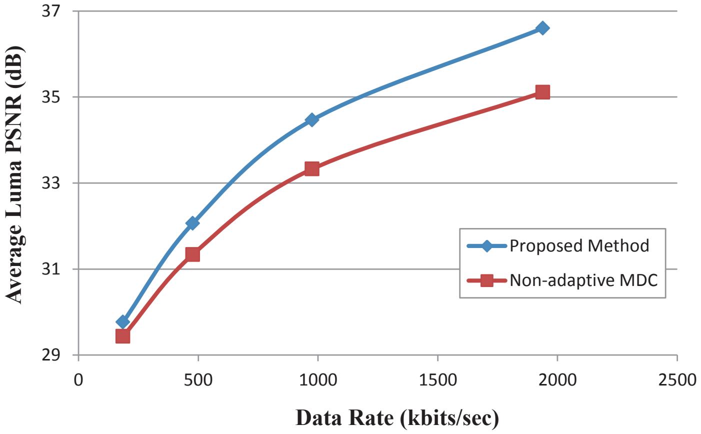
Figure 4.11 and Figure 4.12 show the results when both the proposed method and the non-adaptive method have identical packet loss positions. Figure 4.11 shows that the *“building”* area and the *“bridge”* area of our proposed method are much sharper than the non-adaptive method and the overall visual quality is much better. Figure 4.12 shows results for the *“Mother”* sequence, which is a medium-motion sequence. Note that the *“daughter’s face”* area of our proposed method is clearer than the non-adaptive method.



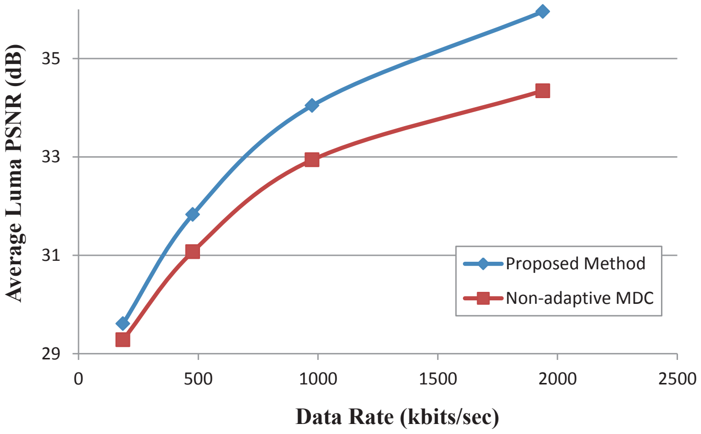
(a) Packet Loss = 5%.



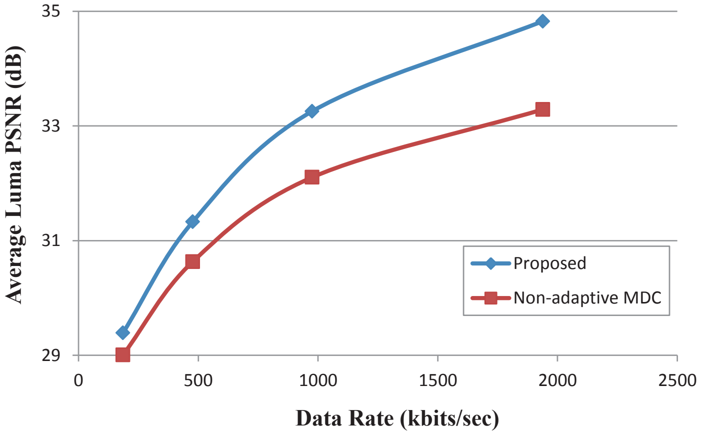
(b) Packet Loss = 10%.



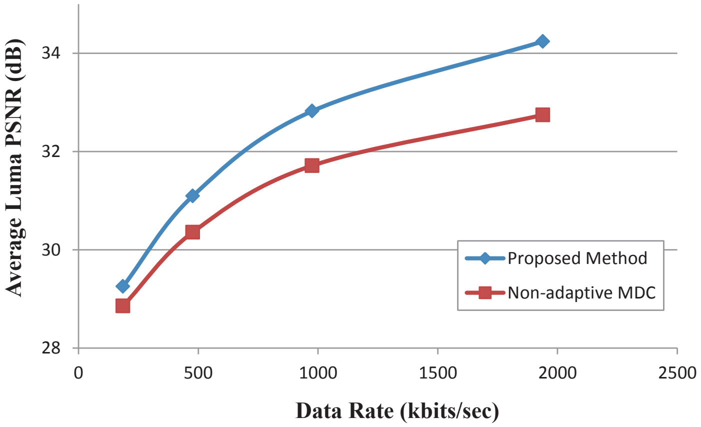
(c) Packet Loss = 15%.



(d) Packet Loss = 20%.

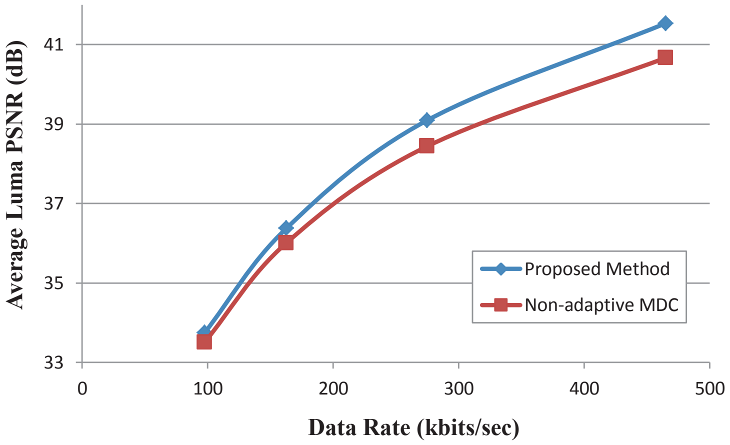


(e) Packet Loss = 25%.

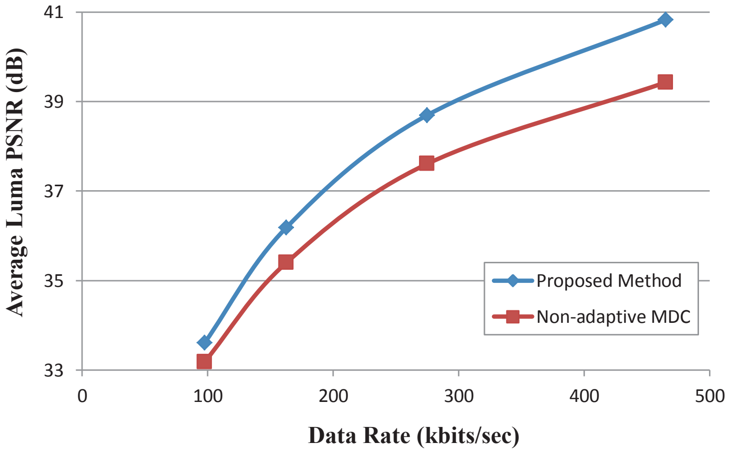


(f) Packet Loss = 30%.

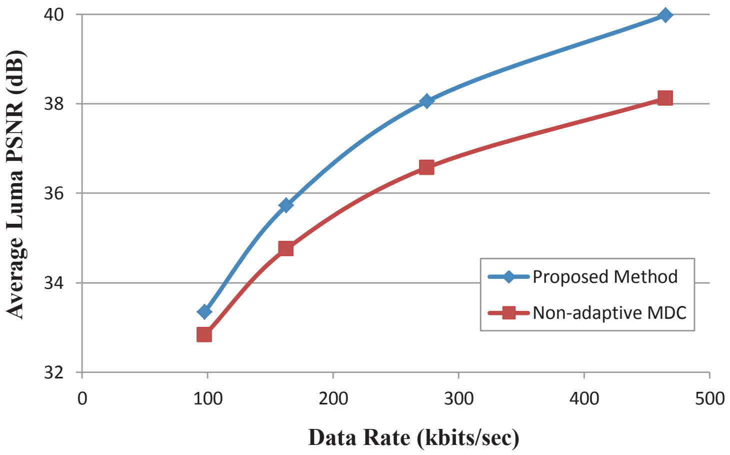
Fig. 4.2. Packet Loss Performances Comparison for the *"Bridge"* Sequence (Non-adaptive MDC denotes the method described in Chapter 3).



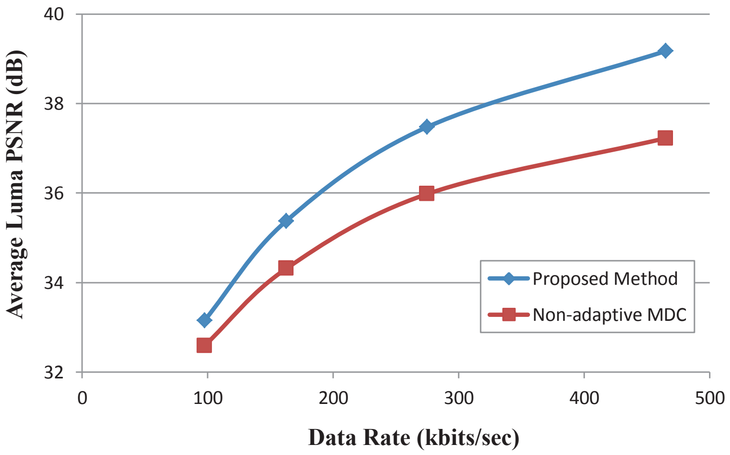
(a) Packet Loss = 5%.



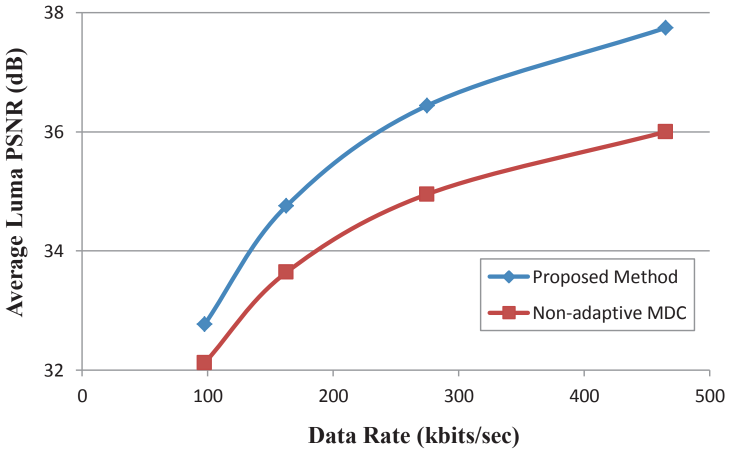
(b) Packet Loss = 10%.



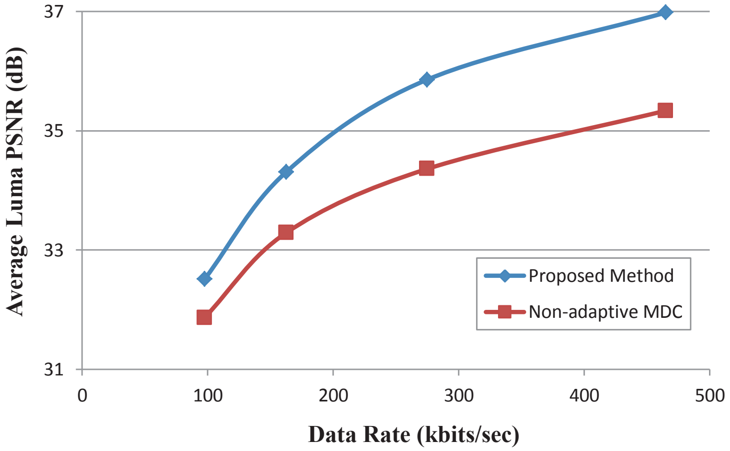
(c) Packet Loss = 15%.



(d) Packet Loss = 20%.

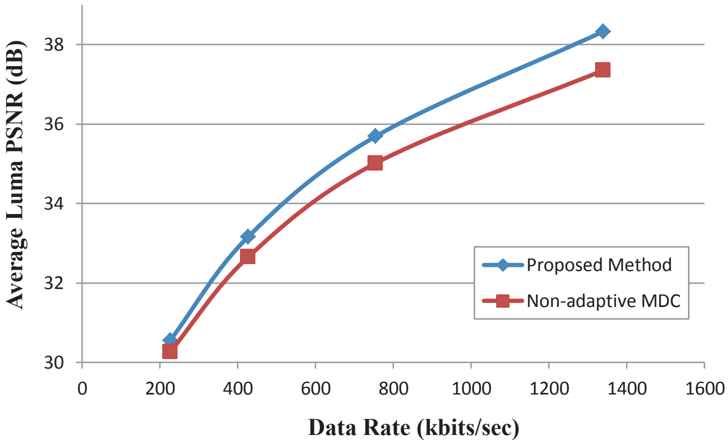


(e) Packet Loss = 25%.

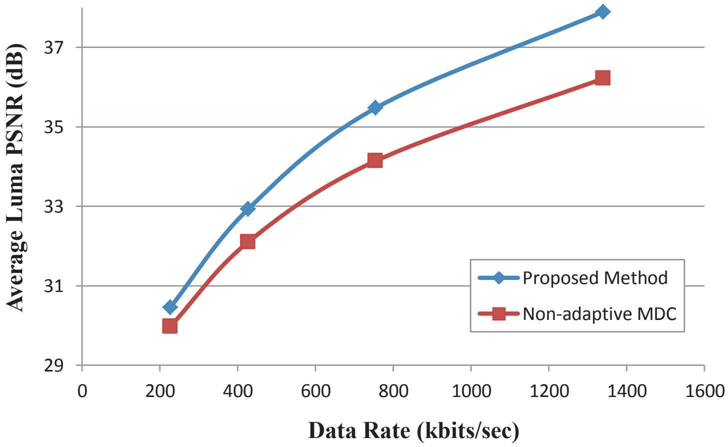


(f) Packet Loss = 30%.

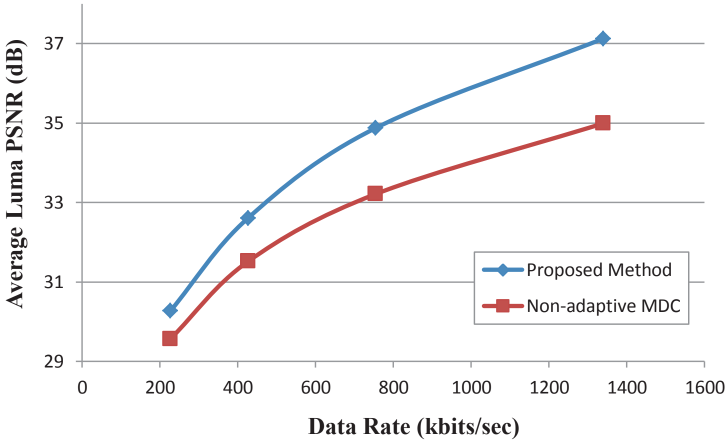
Fig. 4.3. Packet Loss Performances Comparison for the *"Akiyo"* Sequence (Non-adaptive MDC denotes the method described in Chapter 3).



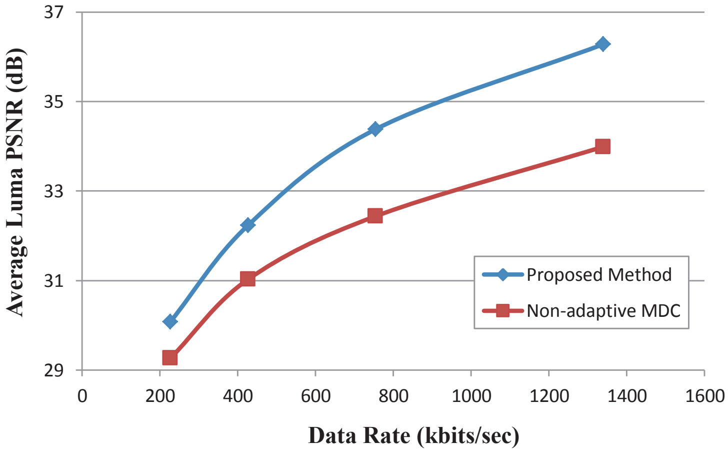
(a) Packet Loss = 5%.



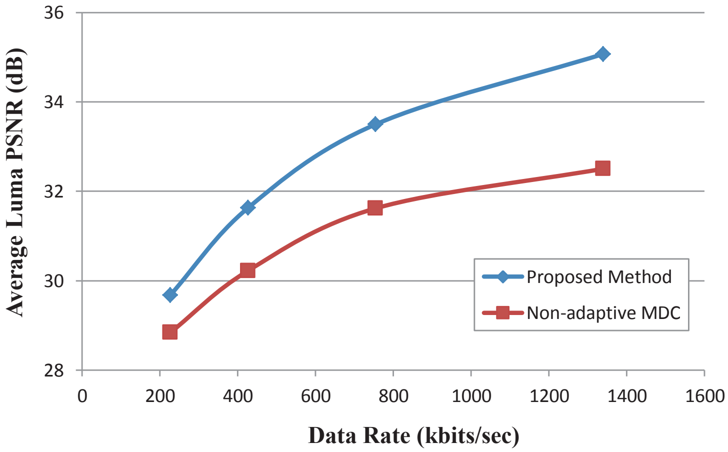
(b) Packet Loss = 10%.



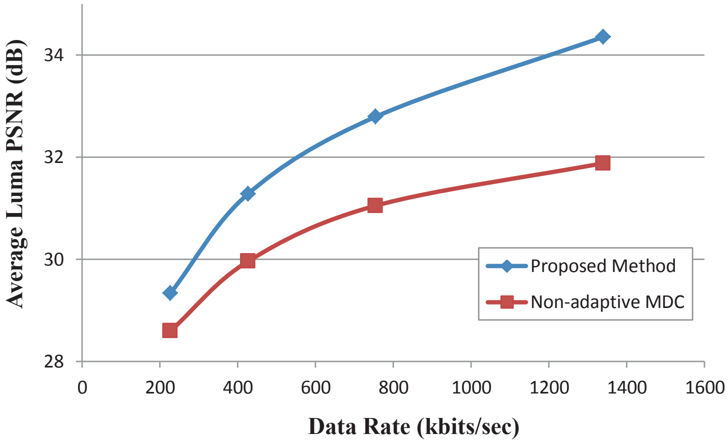
(c) Packet Loss = 15%.



(d) Packet Loss = 20%.

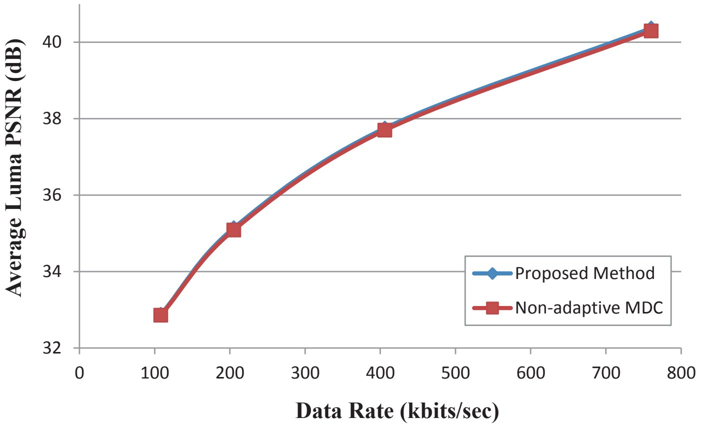


(e) Packet Loss = 25%.

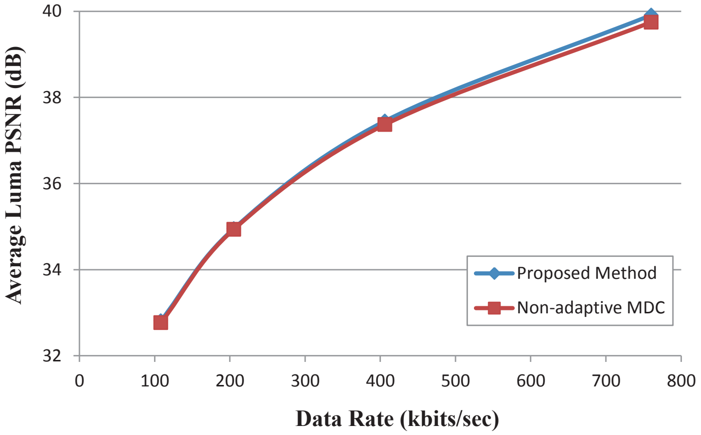


(f) Packet Loss = 30%.

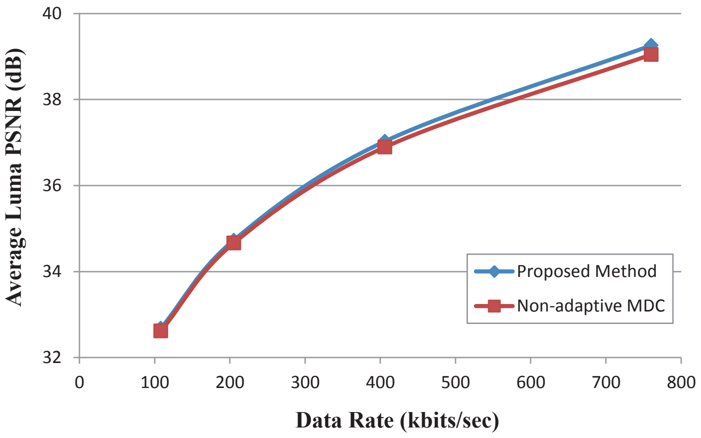
Fig. 4.4. Packet Loss Performances Comparison for the *"Container"* Sequence (Non-adaptive MDC denotes the method described in Chapter 3).



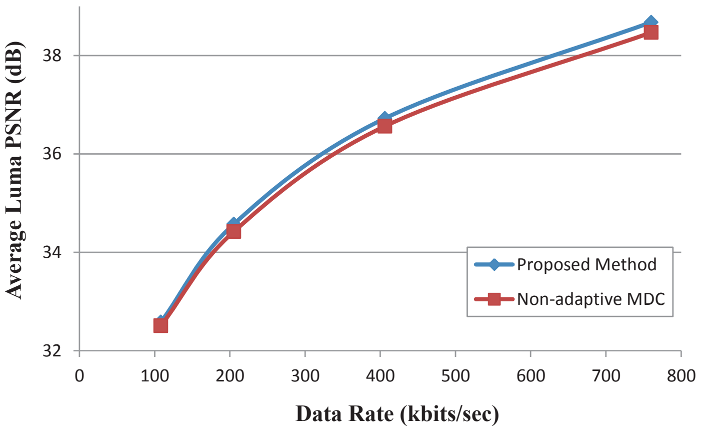
(a) Packet Loss = 5%.



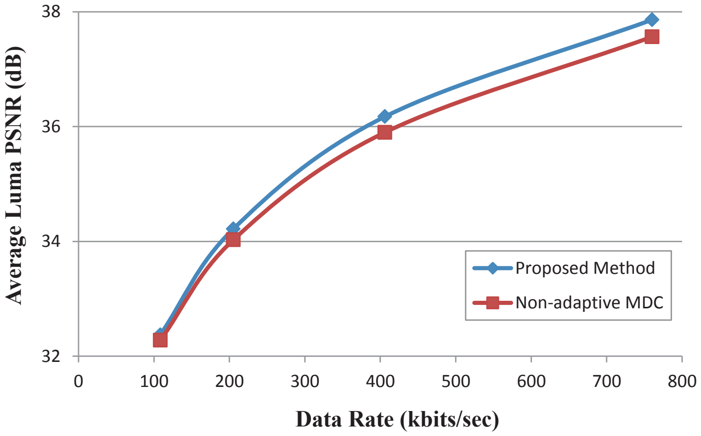
(b) Packet Loss = 10%.



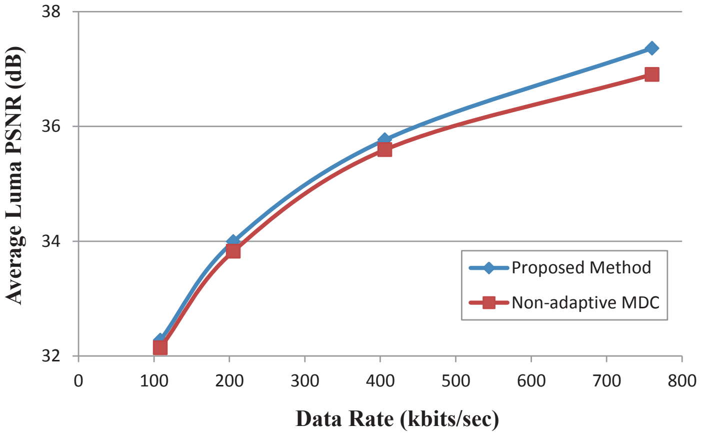
(c) Packet Loss = 15%.



(d) Packet Loss = 20%.

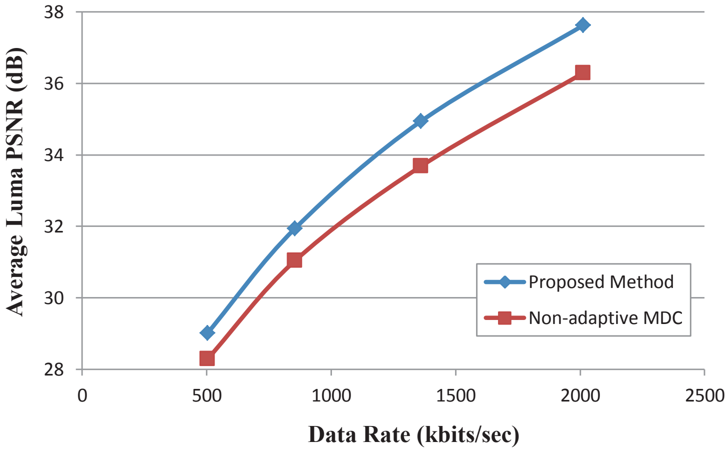


(e) Packet Loss = 25%.

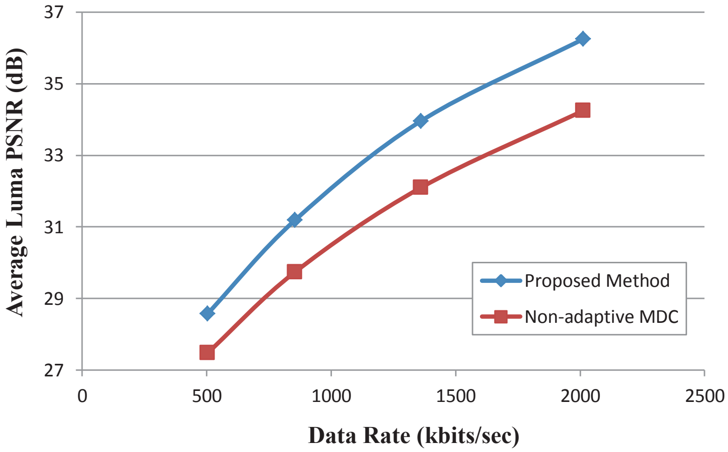


(f) Packet Loss = 30%.

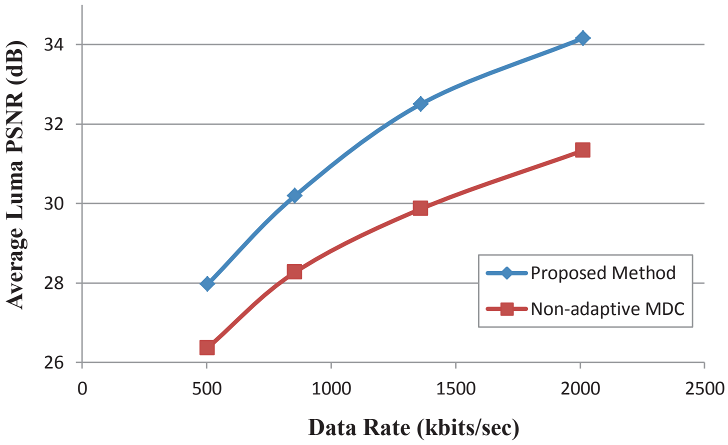
Fig. 4.5. Packet Loss Performances Comparison for the *"Mother"* Sequence (Non-adaptive MDC denotes the method described in Chapter 3).



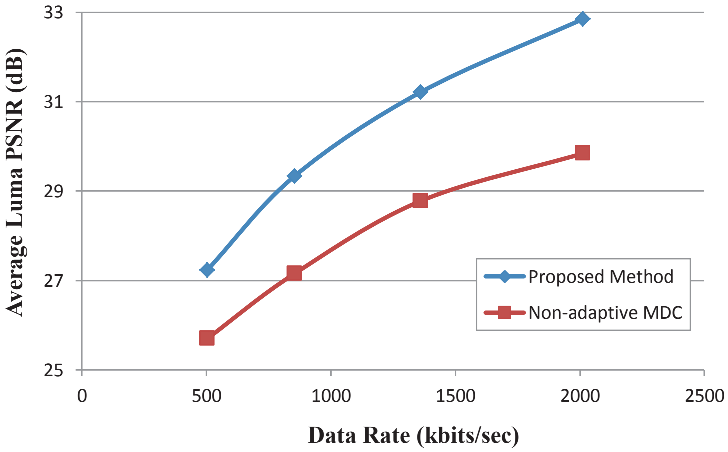
(a) Packet Loss = 5%.



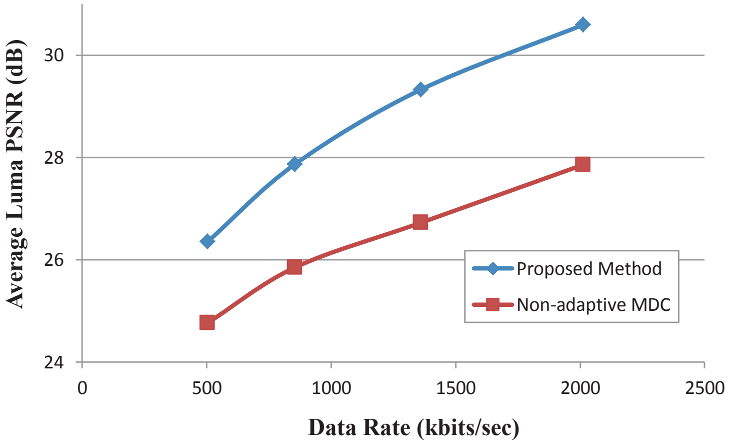
(b) Packet Loss = 10%.



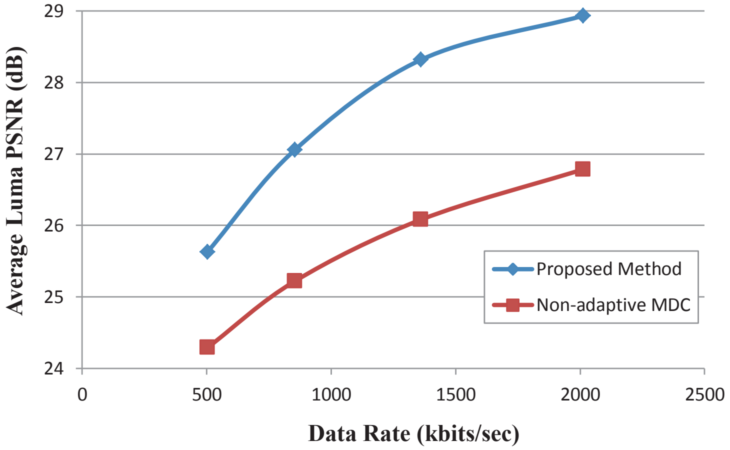
(c) Packet Loss = 15%.



(d) Packet Loss = 20%.

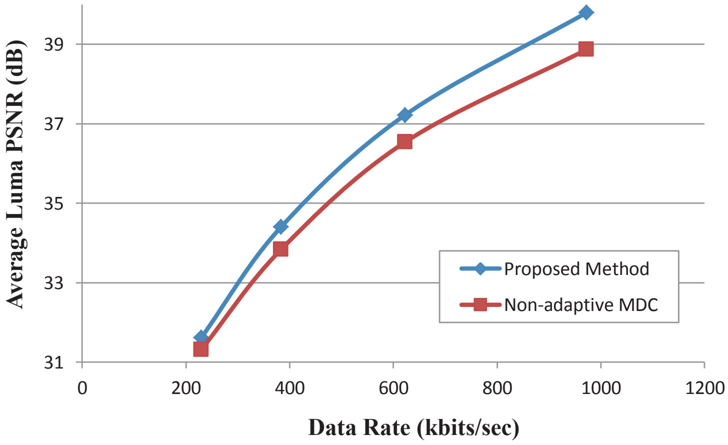


(e) Packet Loss = 25%.

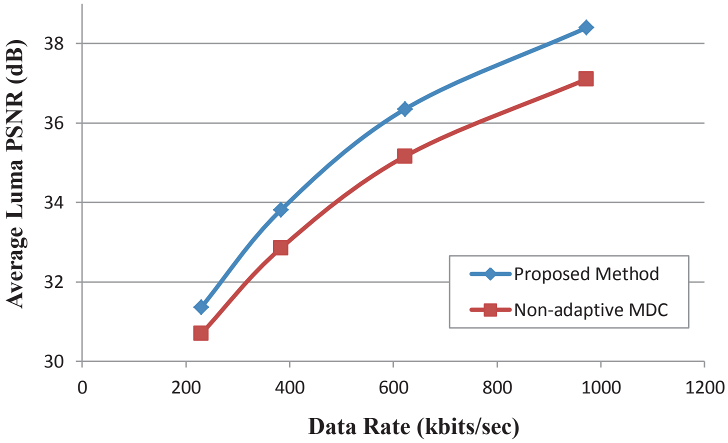


(f) Packet Loss = 30%.

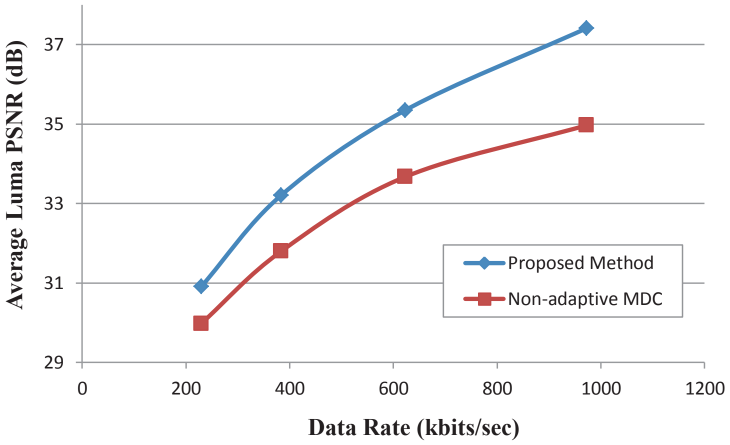
Fig. 4.6. Packet Loss Performances Comparison for the *"Paris"* Sequence (Non-adaptive MDC denotes the method described in Chapter 3).



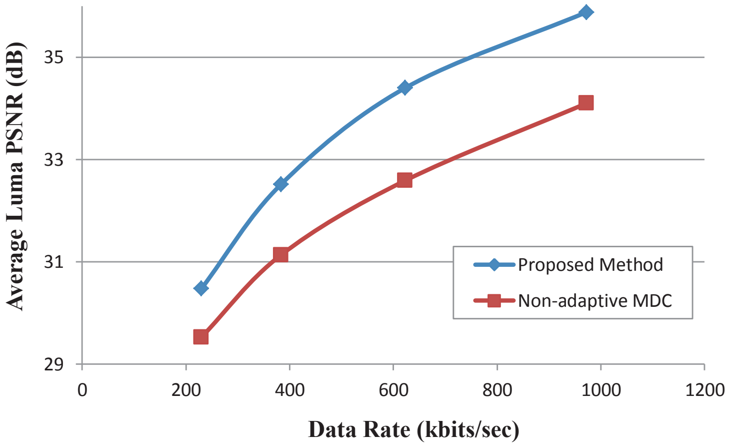
(a) Packet Loss = 5%.



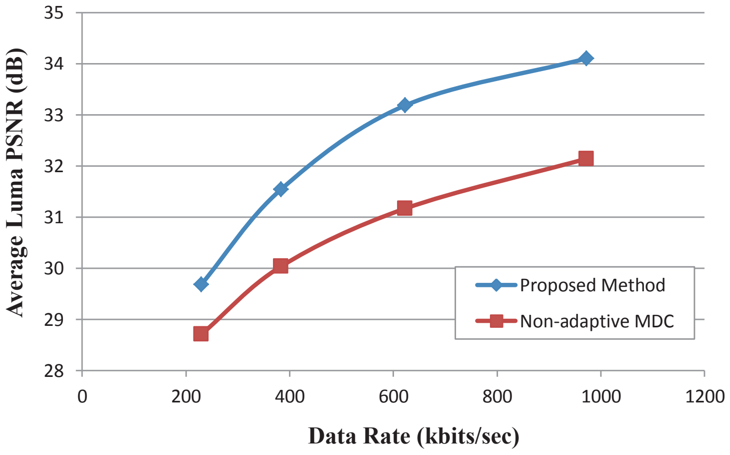
(b) Packet Loss = 10%.



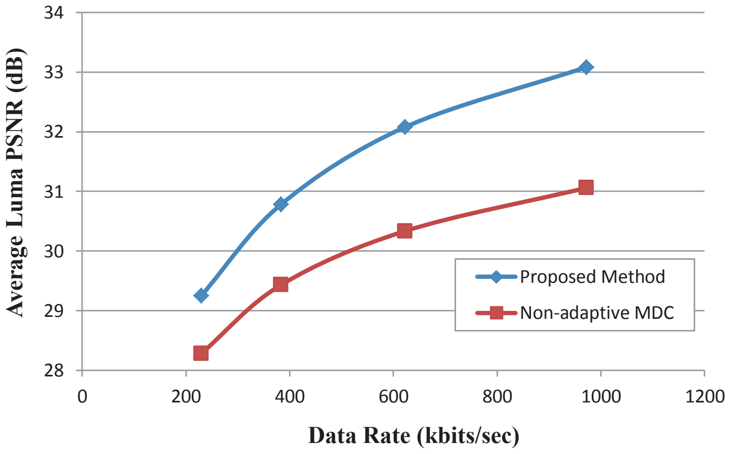
(c) Packet Loss = 15%.



(d) Packet Loss = 20%.

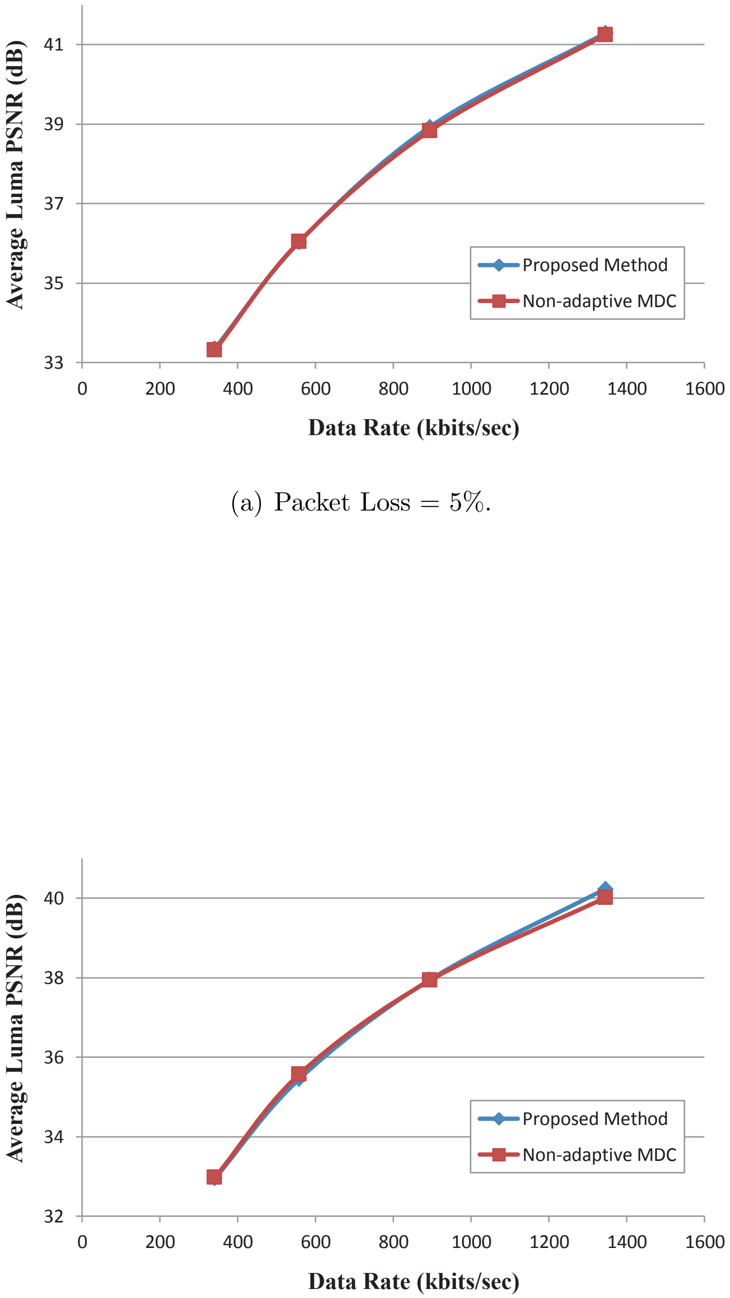


(e) Packet Loss = 25%.



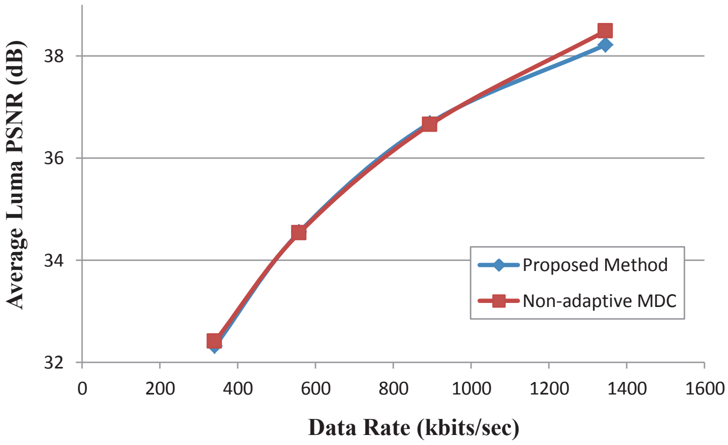
(f) Packet Loss = 30%.

Fig. 4.7. Packet Loss Performances Comparison for the *"News"* Sequence (Non-adaptive MDC denotes the method described in Chapter 3).

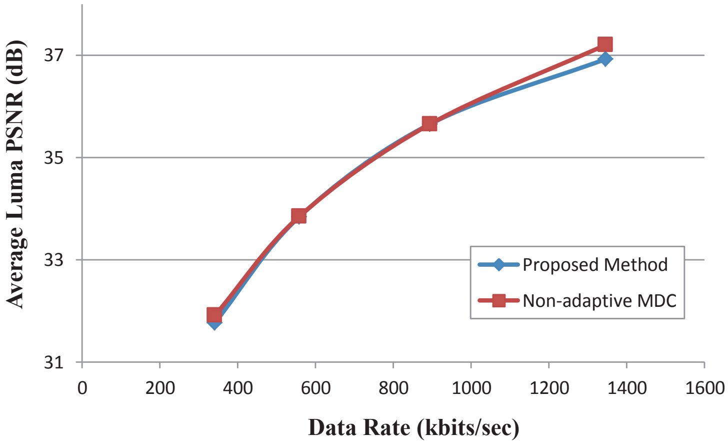


(a) Packet Loss = 5%.

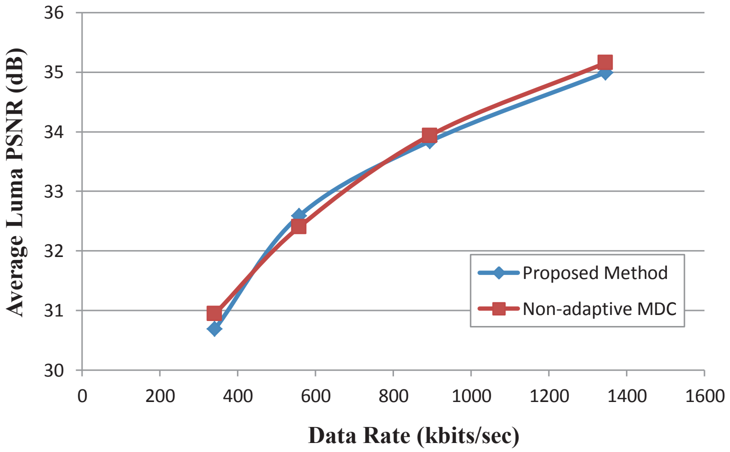
(b) Packet Loss = 10%.



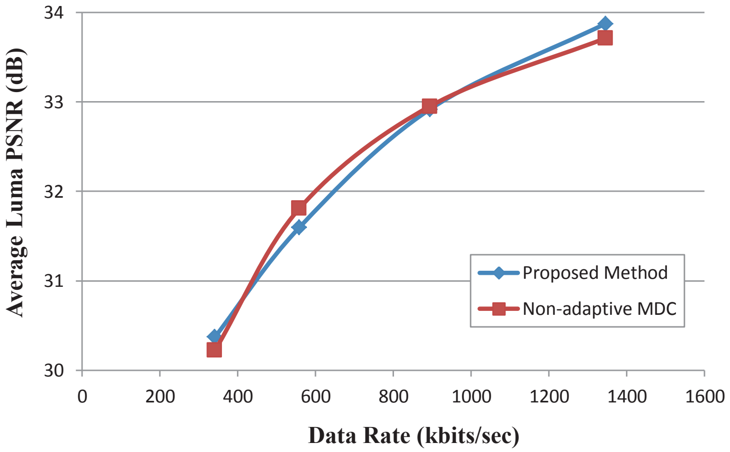
(c) Packet Loss = 15%.



(d) Packet Loss = 20%.

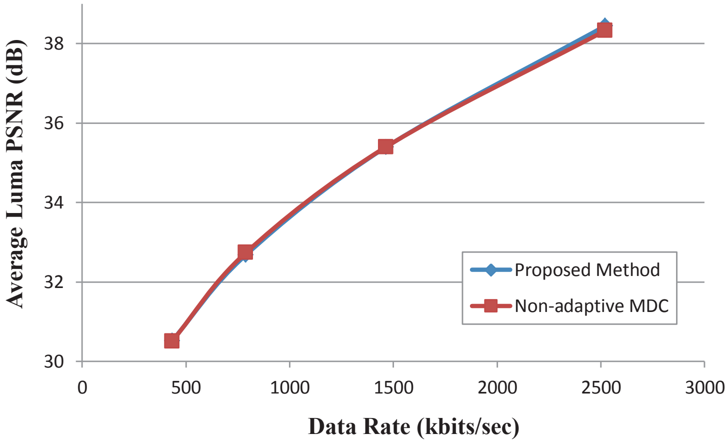


(e) Packet Loss = 25%.

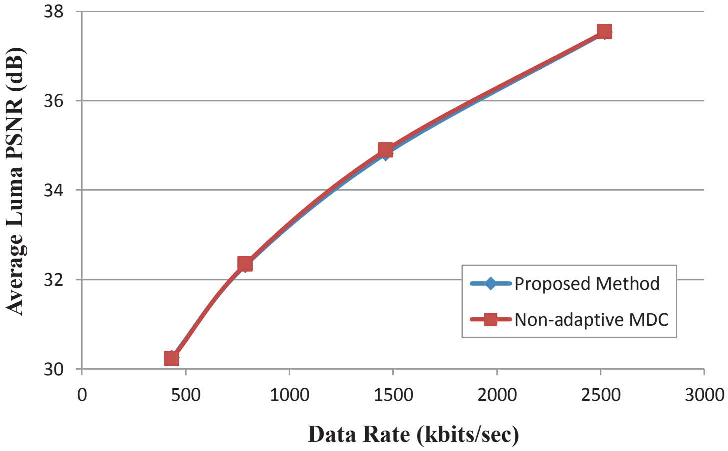


(f) Packet Loss = 30%.

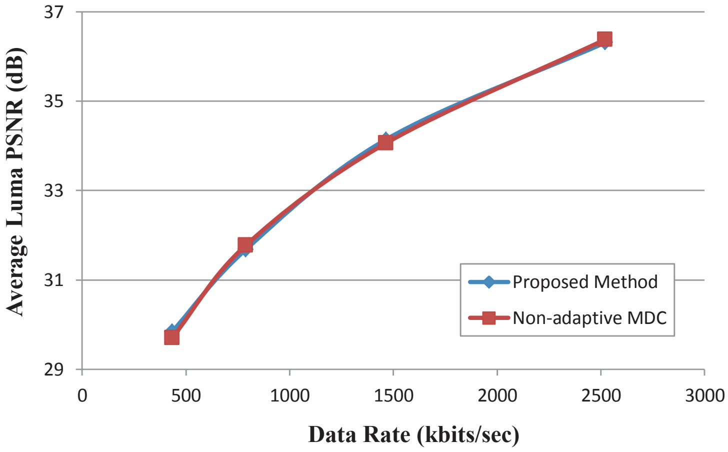
Fig. 4.8. Packet Loss Performances Comparison for the *"Ice"* Sequence (Non-adaptive MDC denotes the method described in Chapter 3).



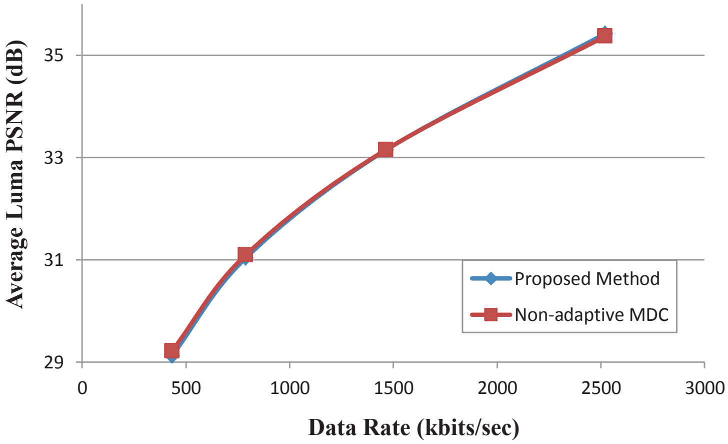
(a) Packet Loss = 5%.



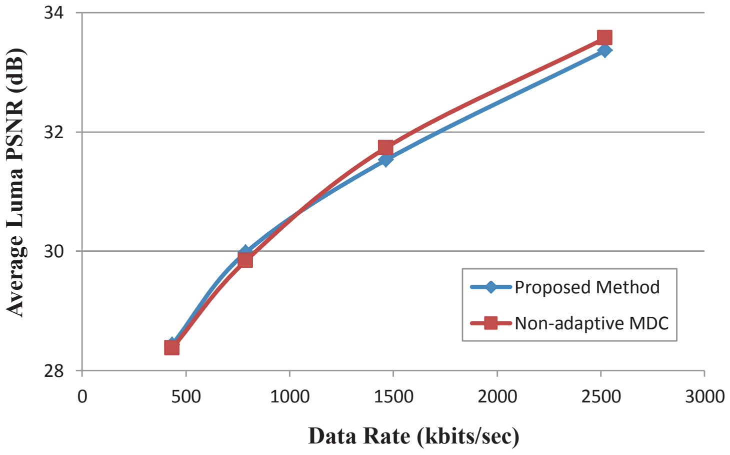
(b) Packet Loss = 10%.



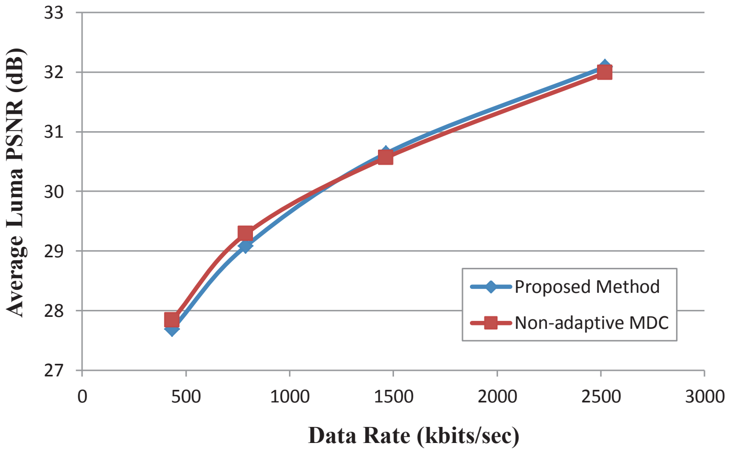
(c) Packet Loss = 15%.



(d) Packet Loss = 20%.

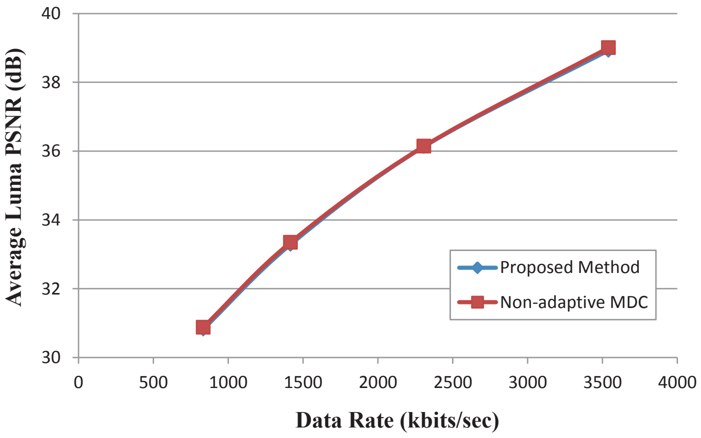


(e) Packet Loss = 25%.

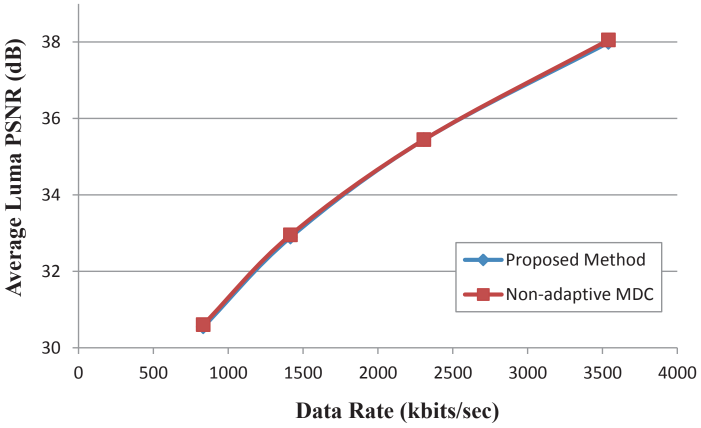


(f) Packet Loss = 30%.

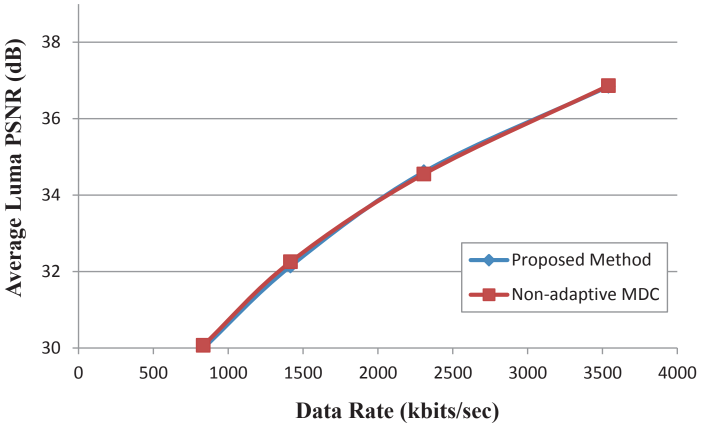
Fig. 4.9. Packet Loss Performances Comparison for the *"Soccer"* Sequence (Non-adaptive MDC denotes the method described in Chapter 3).



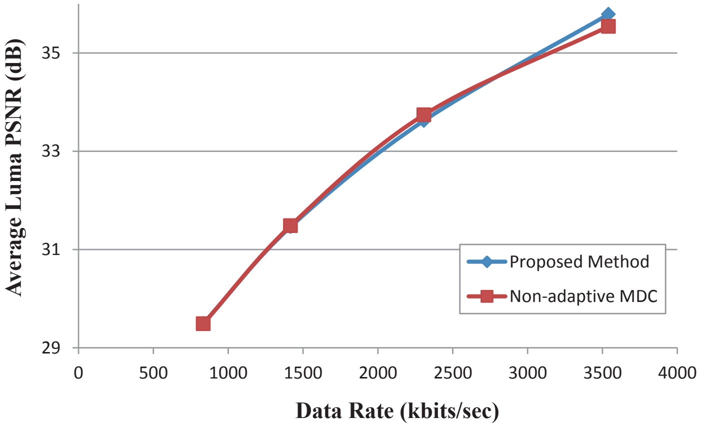
(a) Packet Loss = 5%.



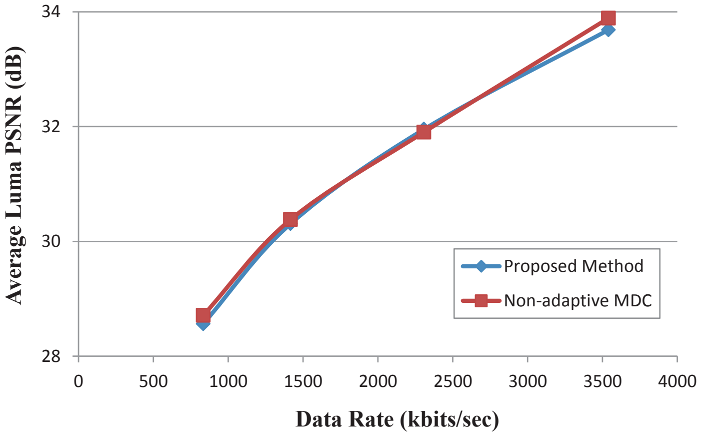
(b) Packet Loss = 10%.



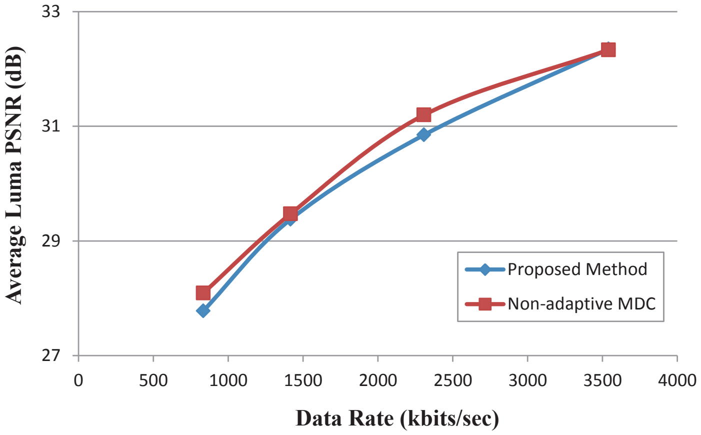
(c) Packet Loss = 15%.



(d) Packet Loss = 20%.

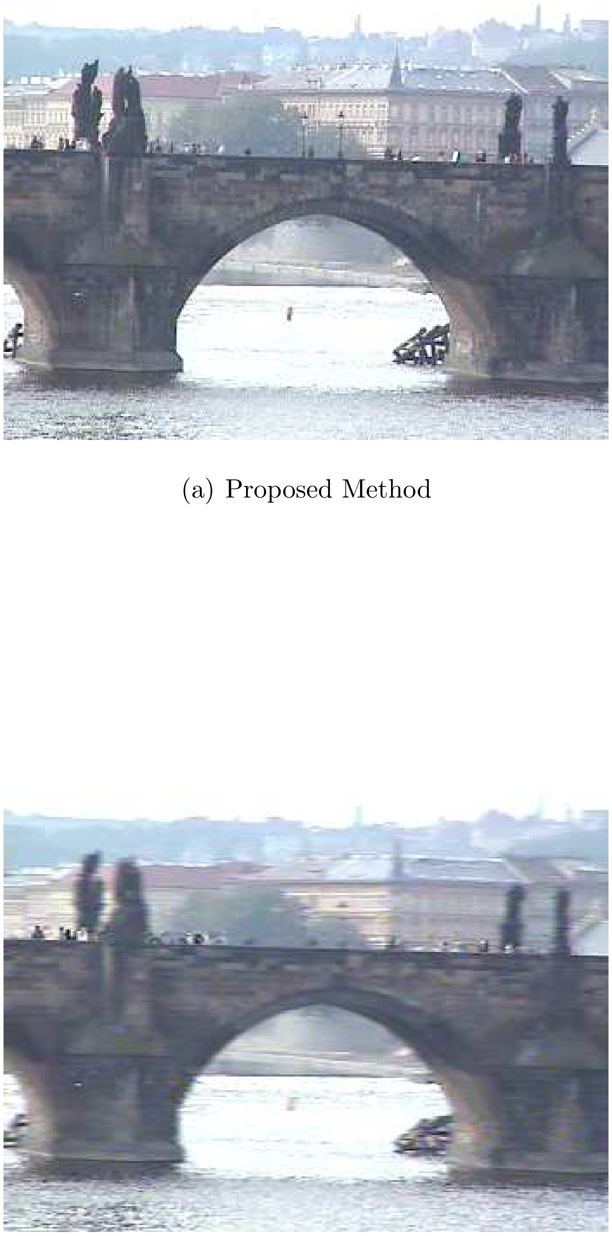


(e) Packet Loss = 25%.



(f) Packet Loss = 30%.

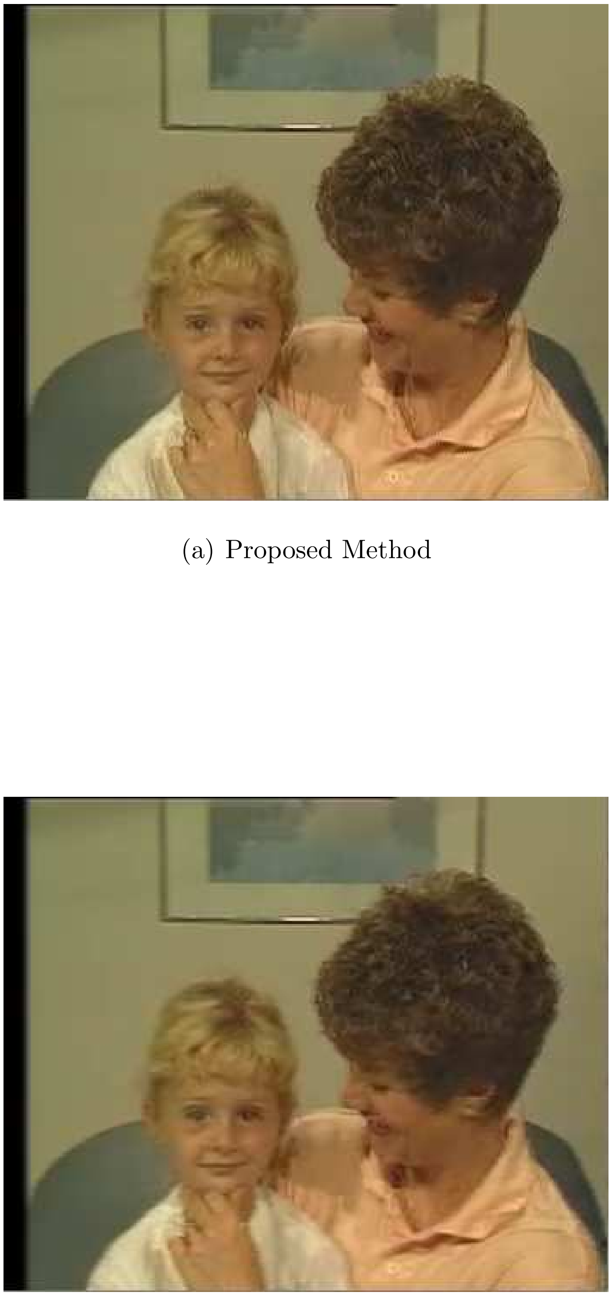
Fig. 4.10. Packet Loss Performances Comparison for the *"Football"* Sequence (Non-adaptive MDC denotes the method described in Chapter 3).



(a) Proposed Method

(b) Non-Adaptive Method

Fig. 4.11. Packet Loss Performances Comparison for the *“Bridge”* Sequence with Identical Packet Loss Positions.



(a) Proposed Method

(b) Non-Adaptive Method

Fig. 4.12. Packet Loss Performances Comparison for the *"Mother"* Sequence with Identical Packet Loss Positions.

Therefore, it appears that our proposed method has an obvious improvement over the non-adaptive method for low-motion sequences, slight improvement for medium-motion sequences, and no improvement for high-motion sequences. This is because, as described in Chapter 3, we use temporal correlation only when both the spatially correlated descriptions of Odd frames or Even frames are lost; otherwise, we use spatial correlation to recover the lost data. Therefore the previous method works well for high-motion sequences but is weak in terms of error robustness for low-motion sequences. And our proposed adaptive method can improve our previous work by adapting to any type of sequences.

4.4 Conclusion

In this chapter, we propose an adaptable spatial-temporal error concealment method for MDC based on error tracking. Experimental results demonstrate the efficacy of the proposed method. Our proposed method improves the non-adaptive method described in Chapter 3 by adapting to any type of sequences.

In this chapter, we choose the optimal error concealment method at the frame level. Usually a frame can contain high motion and low motion at the same time. Hence it is not advantageous to choose the error concealment method at frame level. In the next chapter we describe an adaptive error concealment scheme that chooses between temporal concealment and spatial concealment at the macroblock level.

5. MACROBLOCK-LEVEL ADAPTIVE ERROR CONCEALMENT METHODS FOR MDC

Both methods discussed in Chapter 3 and Chapter 4 describe error concealment at the frame level. However, a frame may contain high motion and low motion at the same time. Therefore, it is advantageous to choose the error concealment method on the macroblock (MB) level. In this chapter we present two MB-level adaptive spatial-temporal error concealment methods with the same MDC partition scheme as the other two methods described in Chapter 3 and Chapter 4. This chapter is organized as follows: Section 5.1 provides a detailed description of the proposed method [101]. Section 5.2 describes the error concealment schemes for the proposed method. Experimental results and discussions are provided in Section 5.3. And we conclude this chapter in Section 5.4.

5.1 MB-Level Error Concealment Methods Based on Foreground-Background Mapping and Distortion Mapping

When the four descriptions generated by the MDC splitter are correctly received, each description can be decoded independently with the final reconstructed video as a the combination of the four decoded sub-videos. However, when packet loss occurs during transmission, joint decoding is done. A frame may contain low motion and high motion at the same time; therefore, it is advantageous to choose the error concealment method on the MB level. To investigate this problem, we label each macroblock with a spatial concealment flag (S) or a temporal concealment flag (T) before encoding. The concealment flags are transmitted as side information to the decoder. In this section, we propose two mappings to label the macroblocks: foreground-background mapping and distortion mapping.

5.1.1 MB-Level Error Concealment Method Based on Foreground-Background Mapping

In our foreground-background mapping method, global motion is first detected. Let $f_n(i,j)$ be the pixel value at position (i,j) in frame n. The index i,j should be within a macroblock, i.e., $16k \leq i,j \leq 16k+15$ where k is non-negative integer. If the mean squared error between two neighboring frames is within a macroblock $\frac{1}{16^2}\sum\limits_i\sum\limits_j (f_n(i,j)-f_{n-1}(i,j))^2$ is greater than threshold T_1, such a macroblock is assumed to have significant motion. If the percentage of the macroblocks with significant motion is greater than threshold T_2 as illustrated in Equation 5.1, this frame is assumed to have global motion.

$$p\left(\left(\frac{1}{16^2}\sum_i\sum_j (f_n(i,j)-f_{n-1}(i,j))^2\right) > T_1\right) > T_2 \tag{5.1}$$

For a frame with global motion, spatial concealment is a better choice than temporal concealment. Thus for this type of frame, we only use spatial concealment and all the macroblocks within the frame are labeled as "S." In the experiments, we do not use testing sequences with obvious global motion. Because in that case, our proposed method should have the same results as the comparison method which uses spatial concealment as the default method and temporal concealment as the secondary method. To better analyze the performances of our proposed method, we avoid testing sequences with global motion in the experiments.

If no global motion is detected, macroblocks with significant motion are classified as foreground and the others as background. We use spatial concealment for foreground macroblocks and temporal concealment for background macroblocks, as illustrated in Equation 5.2.

$$flag = \begin{cases} S & \left(\frac{1}{16^2}\sum\limits_i\sum\limits_j (f_n(i,j)-f_{n-1}(i,j))^2\right) > T_1 \\ T & \qquad\qquad others \end{cases} \tag{5.2}$$

If T_1 is too small, many background macroblocks are classified as foreground, which could blur the image. If T_1 is too large, many foreground macroblocks are classified as background, which could induce severe mismatch artifact. It is not feasible to find a "perfect" threshold to differentiate foreground macroblocks and background macroblocks. To approach the "optimal" threshold, we rank the macroblocks in a descending order based on the histogram of the mean squared error $\frac{1}{16^2}\sum_i\sum_j(f_n(i,j)-f_{n-1}(i,j))^2$ and take the top part of macroblocks as the foreground with a certain percentage which is sequence depended in our experiments.

5.1.2 MB-Level Error Concealment Method Based on Distortion Mapping

In our distortion mapping method, both spatial and temporal concealment are done on each "authentic" macroblock (not corrupted) before encoding and the method with smallest distortion is used as the concealment method, as illustrated in Equation 5.3.

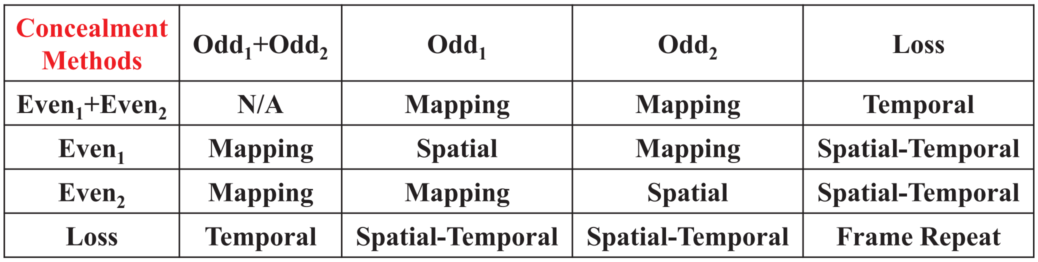
$$flag = \begin{cases} S & dis(spatial) < dis(temporal) \\ T & others \end{cases} \tag{5.3}$$

Each concealment flag in Equation 5.2 and 5.3 need one bit to be coded. In our experiments, we use testing sequences with original resolutions of CIF (352 × 288) at 30 frames/sec. So the additional data rate caused by side information for each sequence is $\frac{352\times288}{16\times16}\times30\times10^{-3}=11.88kbps$, which is negligible compared with the encoding data rate.

5.2 Error Concealment Schemes

When the four descriptions generated by the MDC splitter are correctly received, each description can be decoded independently with the final reconstructed video a the combination of the four decoded sub-videos. However, when packet loss occurs during transmission, joint decoding is done.

Table 5.1 shows the error concealment schemes in each packet receiving scenario, where the first row lists receiving scenarios of Odd frames and the first column lists



Concealment Methods	Odd$_1$+Odd$_2$	Odd$_1$	Odd$_2$	Loss
Even$_1$+Even$_2$	N/A	Mapping	Mapping	Temporal
Even$_1$	Mapping	Spatial	Mapping	Spatial-Temporal
Even$_2$	Mapping	Mapping	Spatial	Spatial-Temporal
Loss	Temporal	Spatial-Temporal	Spatial-Temporal	Frame Repeat

Table 5.1
Error Concealment Schemes.

receiving scenarios of Even frames. "Odd$_1$+Odd$_2$" means both Odd$_1$ and Odd$_2$ sequences are received, and "Loss" in the first row denotes that neither Odd$_1$ nor Odd$_2$ is received. Similarly, "Even$_1$+Even$_2$" denotes both Even$_1$ and Even$_2$ sequences are received, and "Loss" in the first column denotes neither Even$_1$ nor Even$_2$ is received. "Spatial-Temporal" means spatial concealment is done first then temporal concealment is done. "Mapping" means the concealment method from the mapping table generated by Section 5.1.1 or Section 5.1.2 is used as the error concealment method.

According to Table 5.1, there are three schemes for error concealment. For a lost frame, if both its temporally correlated frame and its spatially correlated frame are received, the concealment method from the mapping table is used for error concealment. If only the temporally(spatially) correlated frame is received, then temporal(spatial) concealment is used for error concealment. If neither the temporally nor the spatially correlated frame is received, we use the previously decoded reference frame for concealment which is named "Frame Repeat" in the table.

For spatial concealment, we use a two-neighbor bilinear filter. For temporal concealment, we use a non-motion compensated method which copies the pixel value of the same position from the frame used for concealment.

5.3 Experimental Results

In this section, we evaluate the packet loss performance of the two proposed methods. The results are compared with our non-adaptive method described in Chapter 3.

All the experiments are implemented by modifying the JVT JM software version 16.1. The testing sequences have original resolutions of CIF (352 × 288) at 30 frames/sec with 200-frame length. Thus each description has 100 subframes at 15 frames/sec. The coding structure is "IPPP...", with an I-frame refreshment every 15 subframes in each description. The quantization parameters for I frames and P frames are 20, 24, 28 and 32.

Two testing sequences are used in our experiments: the *"Paris"* sequence and the *"News"* sequence. Experimental results are obtained by averaging 100 channel transmission simulations using the Gilbert model. As noted in Section 3.3, when P_B is small, L_B is large; and vice versa [99]. The parameters for the Gilbert model in our experiments are shown in Table 5.2.

Table 5.2
Gilbert Model Parameters for Various Packet Loss Rates

Loss Rate	5%	10%	15%	20%
Burst Length	5	5	4	4

Both the foreground-background mapping and the distortion mapping are based on the downsampled subframe. The four descriptions generated by the MDC splitter share the same mappings. Figure 5.1 shows an example of the two mappings for a subframe of the *"Paris"* sequence. In the mappings, macroblocks in black stand for low-motion macroblocks and temporal concealment is used for these macroblocks. Macroblocks in white denote high-motion macroblocks and spatial concealment is used for these macroblocks. In this example, the *"shelf"* area and the *"desk"* area have no motion and the motion of the man and the woman are not significant, while the *"woman's moving hands"* area has significant motion. For our adaptive method based on foreground-background mapping, all the macroblocks except those in the *"woman's moving hands"* area are mapped to background according to our thresholds. For our adaptive method based on distortion mapping, macroblocks in the *"man's moving head"* area, the *"woman's moving head"* area, the *"man's moving hands"* area and the *"woman's moving hands"* area are mapped to foreground.

Figure 5.2 shows the results when the two adaptive methods and the non-adaptive method have identical packet loss positions. Note that the *"shelf"* area and the *"desk"* area in (a) and (b) are much sharper than (c). This is because all the macroblocks in these two areas are classified as low-motion macroblock which uses temporal concealment and maintains the sharpness. However, for the non-adaptive method, these macroblocks are concealed by spatial concealment when they are lost, and the image is blurred. Most of the macroblocks in the *"woman's moving hands"* area in (a) and (b) are classified as high-motion macroblock, and spatial concealment is used for these macroblocks. Thus the quality of this area in (a) and (b) are similar with (c) which uses spatial concealment as the default concealment method and temporal concealment as the secondary method. In addition, the quality of the two adaptive methods are similar.

Figure 5.3 and Figure 5.4 show packet loss performances of the *"Paris"* and the *"News"* sequences respectively. Packet loss rates are from 5% to 20%. The performance of the two proposed adaptive methods are similar and are much better than the non-adaptive method. In addition, when the packet loss rate increases, both of the proposed adaptive methods outperform the non-adaptive method in an increasing rate. For the *"Paris"* sequence, when packet loss rate reaches 20%, our proposed adaptive methods have a gain up to 3.3 dB in terms of PSNR. And for the *"News"* sequence, the improvement is as large as 3.7dB.

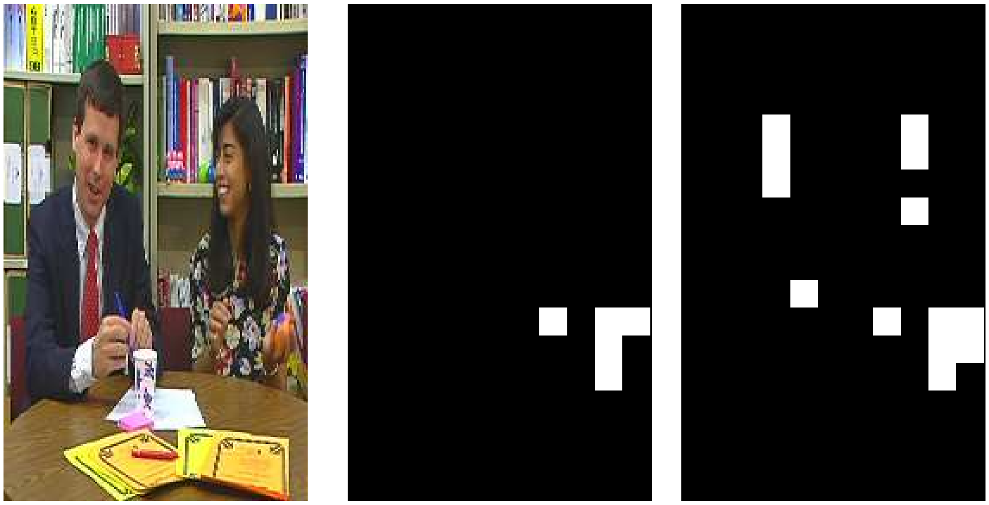
5.4 Conclusion

In this chapter, we proposed two MB-level adaptive error concealment methods for MDC with four descriptions based on foreground-background mapping and distortion mapping respectively. Experimental results demonstrate the efficacy of the proposed methods. Both the subjective quality and objective quality demonstrate these two proposed methods greatly improve the non-adaptive method described in Chapter 3.

But these two methods require the transmission of side information to the decoder for better error concealment. Even though the additional data rate is negligible compared to the total data rate, correct delivery of the side information is very important for these two methods.

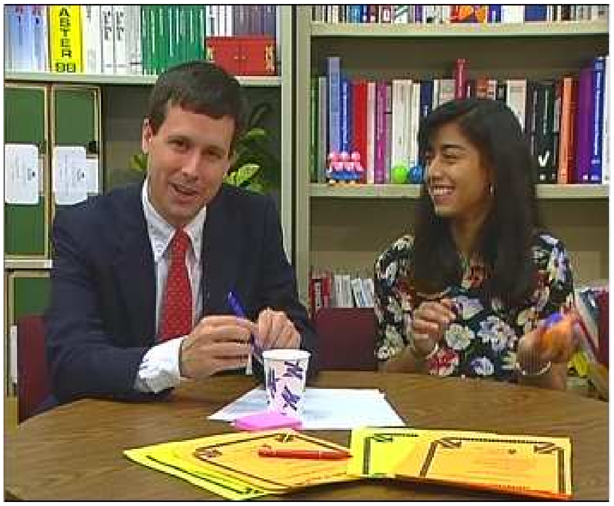
In our MDC partition architecture, each description has an independent prediction loop, and the information used as reference for both the intra-prediction and the inter-prediction are reduced. Therefore, the prediction precision degrades and the residue increases which costs more bits to be coded.

In the next chpater an improved MDC partition architecture for better coding efficiency and error robustness, and an adaptive error concealment method without side information is described.

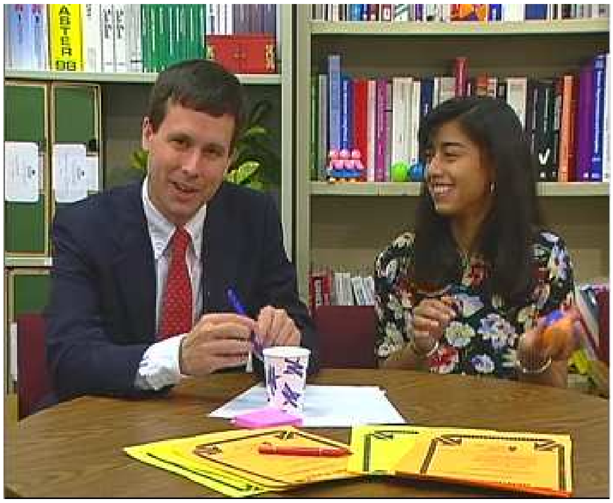


(a) Downsampled Frame from One Description (b) Foreground-Background Mapping (c) Distortion Mapping

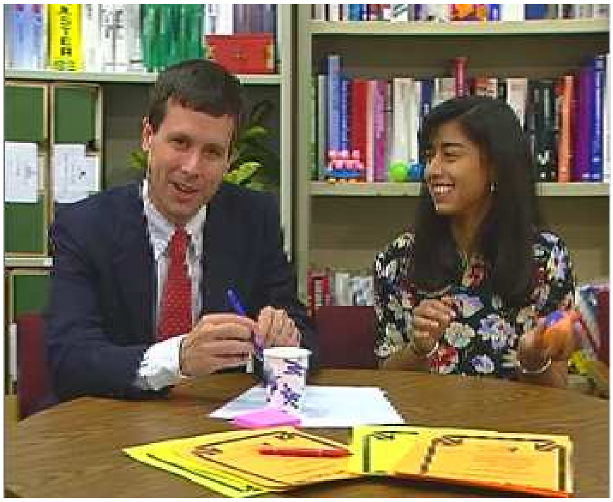
Fig. 5.1. An Example of the Two Mappings for *"Paris"* Sequence.



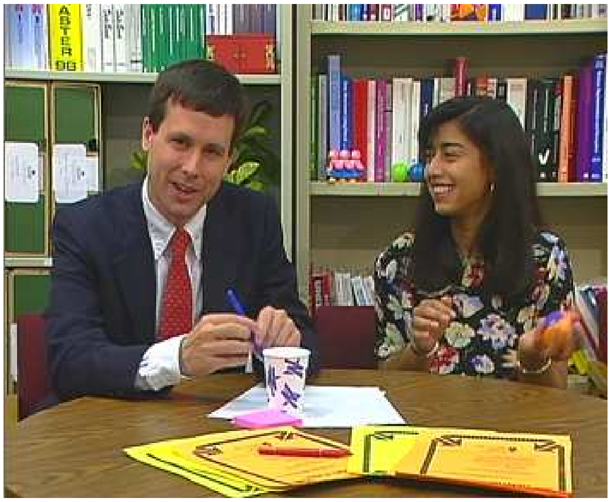
(a) Adaptive Method Based on Distortion Mapping



(b) Adaptive Method Based on Foreground-Background Mapping

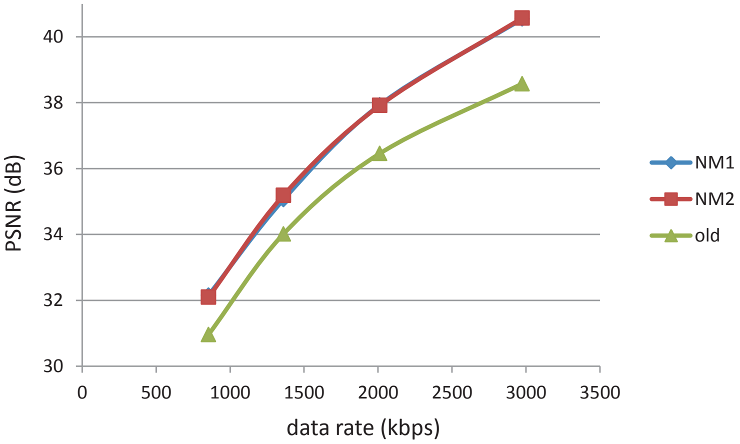


(c) Non-Adaptive Method [95]

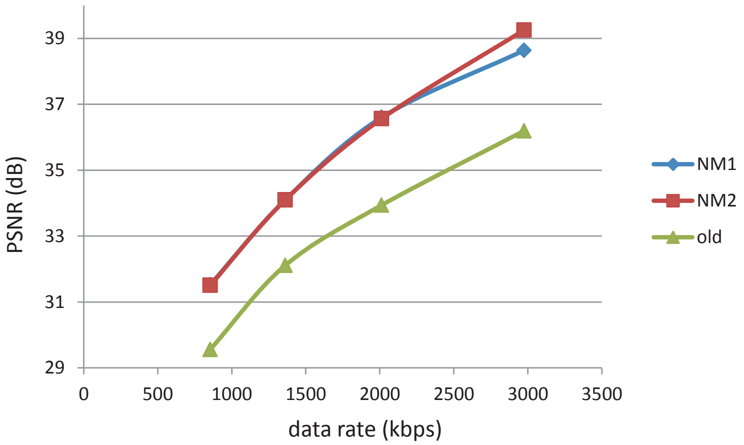


(d) Original Frame

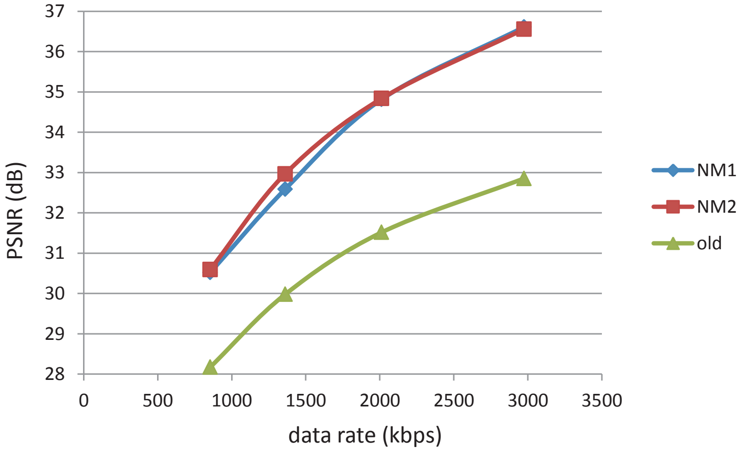
Fig. 5.2. Packet Loss Performances Comparison for *"Paris"* Sequence with Identical Packet Loss Positions.



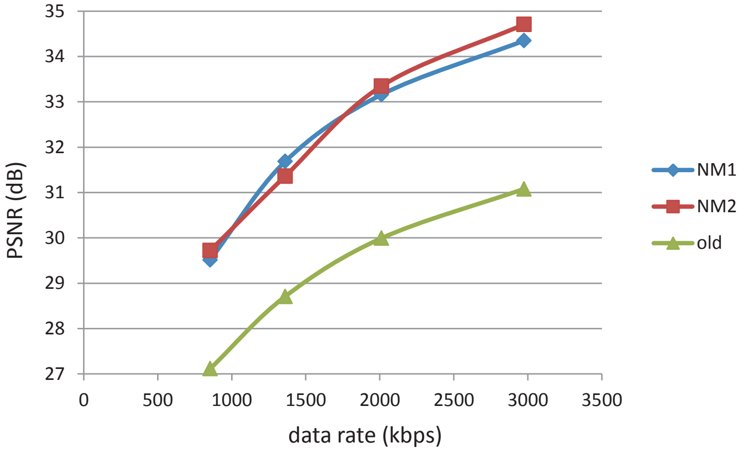
(a) Packet Loss = 5%.



(b) Packet Loss = 10%.

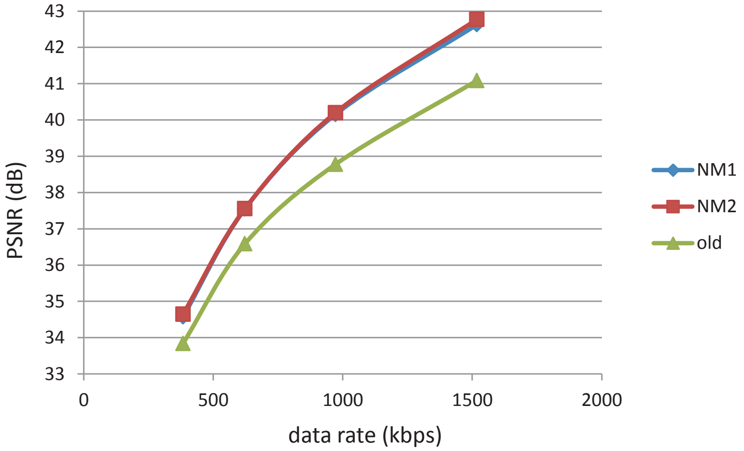


(c) Packet Loss = 15%.

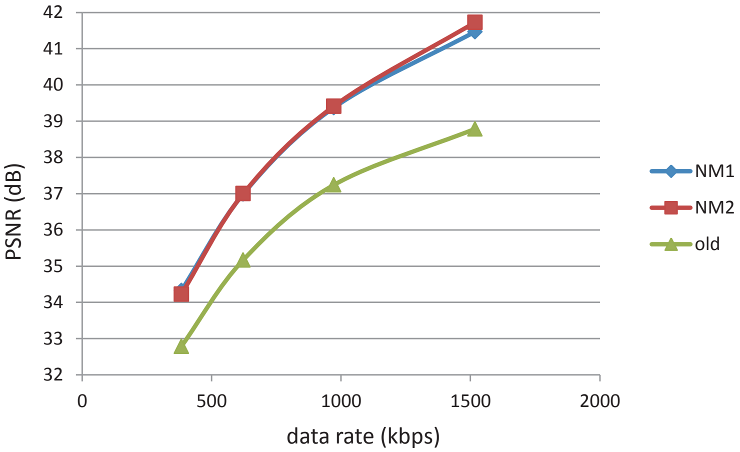


(d) Packet Loss = 20%.

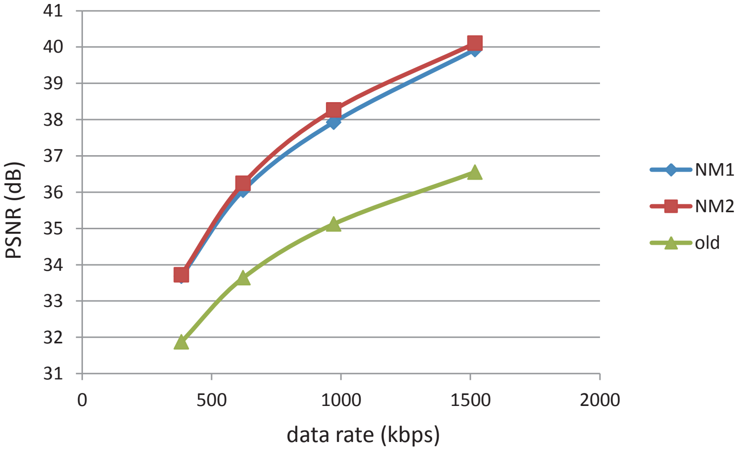
Fig. 5.3. Packet Loss Performances Comparison for *"Paris"* Sequence (NM1 denotes the adaptive method based on foreground-background mapping and NM2 denotes the adaptive method based on distortion mapping).



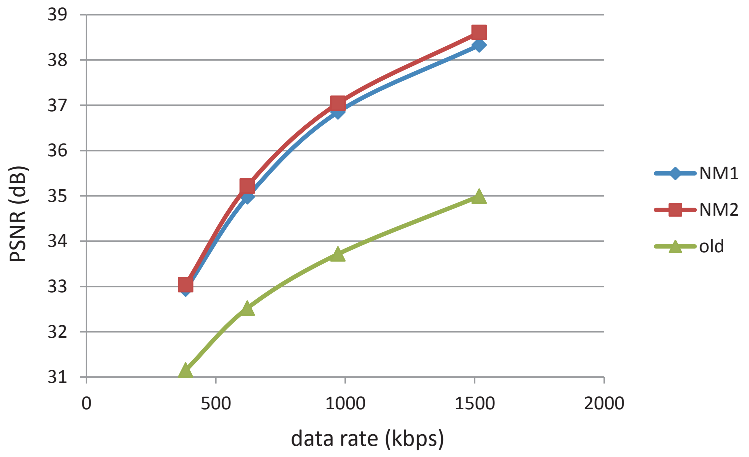
(a) Packet Loss = 5%.



(b) Packet Loss = 10%.



(c) Packet Loss = 15%.



(d) Packet Loss = 20%.

Fig. 5.4. Packet Loss Performances Comparison for *"News"* Sequence (NM1 denotes the adaptive method based on foreground-background mapping and NM2 denotes the adaptive method based on distortion mapping).

6. ADVANCED MULTIPLE DESCRIPTION VIDEO CODING FOR BETTER CODING EFFICIENCY AND ERROR ROBUSTNESS

The work described in Chapters 3, 4 and 5 use the same MDC partition architecture. In that architecture, both temporal separation and spatial separation are outside the prediction loop, so that each description has an independent prediction loop. This architecture is straightforward and a packet loss of one description does not influence the decoding of other descriptions. But this architecture can greatly influence the coding efficiency because the information used as reference for both intra prediction and inter prediction are reduced. As the prediction precision degrades, the residue increases and thus coding efficiency is degraded. In this chapter, we propose an advanced MDC partition architecture in which temporal separation is outside the prediction loop while spatial separation is inside the loop. This chapter is organized as follows: Section 6.1 provides a detailed description of the proposed MDC architecture. Section 6.2 describes the error concealment schemes for the proposed method. Experimental results and discussions are provided in Section 6.3. And we conclude this chapter in Section 6.4.

6.1 MDC Partition Architecture

The new MDC partition consists of two parts: temporal separation and spatial separation. The original video sequence is first split into two temporally correlated descriptions, which are called the Even and Odd sequences respectively as shown in Figure 6.1. Each of the new sequences has the encoder structure and decoder structure as shown in Figure 6.2 and Figure 6.3. One description is encoded in two descriptions/bitstreams using the encoder shown in Figure 6.2. For I slice, the encoder duplicates the encoded data into each description within the same prediction loop. For P slice, all information except transformed coefficients is duplicated into each description within the same prediction loop, and the coefficients are distributed into two descriptions by residue separation. Figure 6.4 shows the spatial separation. Each 8×8 residue block is split into two 8×8 residue blocks by horizontal downsampling. The “empty” positions are padded with zeros. As a result, Even$_1$ and Odd$_1$ are from odd columns of the residue separation, while Even$_2$ and Odd$_2$ are from even columns. “Even” and “Odd” denotes temporally correlated even numbered frames and odd numbered frames. “1” and “2” denotes spatially correlated odd columns and even columns of residue separation. After transform, quantization, dequantization and inverse transform, the two spatially correlated residue blocks are merged as one 8×8 residue block. Since we only use 4×4 transform, all the padded 4×4 blocks in the newly generated 8×8 residue blocks only require a few bits to encode. At the decoder, after entropy decoding, dequantization and inverse transform, the two spatially correlated residue blocks are merged as one residue block to generate the reconstructed block.

6.2 Error Concealment Schemes

The two spatially correlated descriptions of Odd or Even sequences share the same prediction loop and can be decoded independently from the other two spatially correlated descriptions in the other prediction loop. When the four descriptions are correctly received, the final reconstructed video is the combination of odd and even sequences. However, when packet loss occurs during transmission, joint decoding is done.

Table 6.1 shows the error concealment schemes in each packet receiving scenario, where the first row lists receiving scenarios of Odd frames and the first column lists receiving scenarios of Even frames. “Odd$_1$+Odd$_2$” means both Odd$_1$ and Odd$_2$ sequences are received, and “Loss” in the first row denotes that neither Odd$_1$ nor Odd$_2$

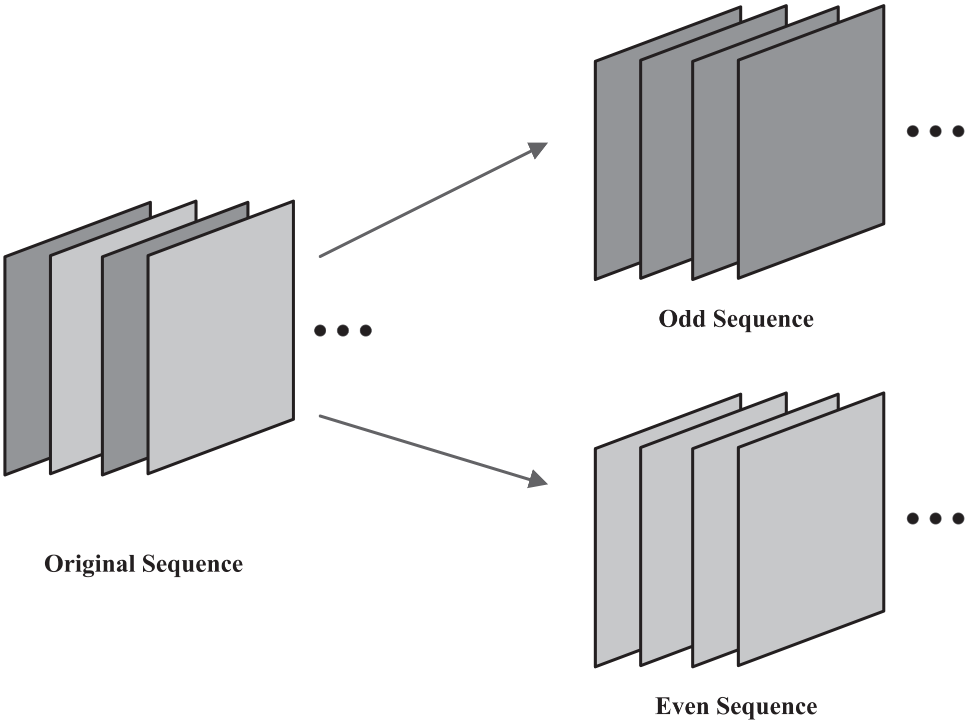


Fig. 6.1. MDC Temporal Separation.

Concealment Methods	Odd$_1$+Odd$_2$	Odd$_1$	Odd$_2$	Loss
Even$_1$+Even$_2$	N/A	Spatial	Spatial	Temporal
Even$_1$	Spatial	Spatial	Spatial	Spatial-Temporal
Even$_2$	Spatial	Spatial	Spatial	Spatial-Temporal
Loss	Temporal	Spatial-Temporal	Spatial-Temporal	Frame Repeat

Table 6.1
Error Concealment Scheme.

is received. Similarly, "Even$_1$+Even$_2$" denotes both Even$_1$ and Even$_2$ sequences are received, and "Loss" in the first column denotes neither Even$_1$ nor Even$_2$ is received. "Spatial-Temporal" means spatial concealment is done first then temporal concealment is done. According to Table 3.1, when one description is received, such as

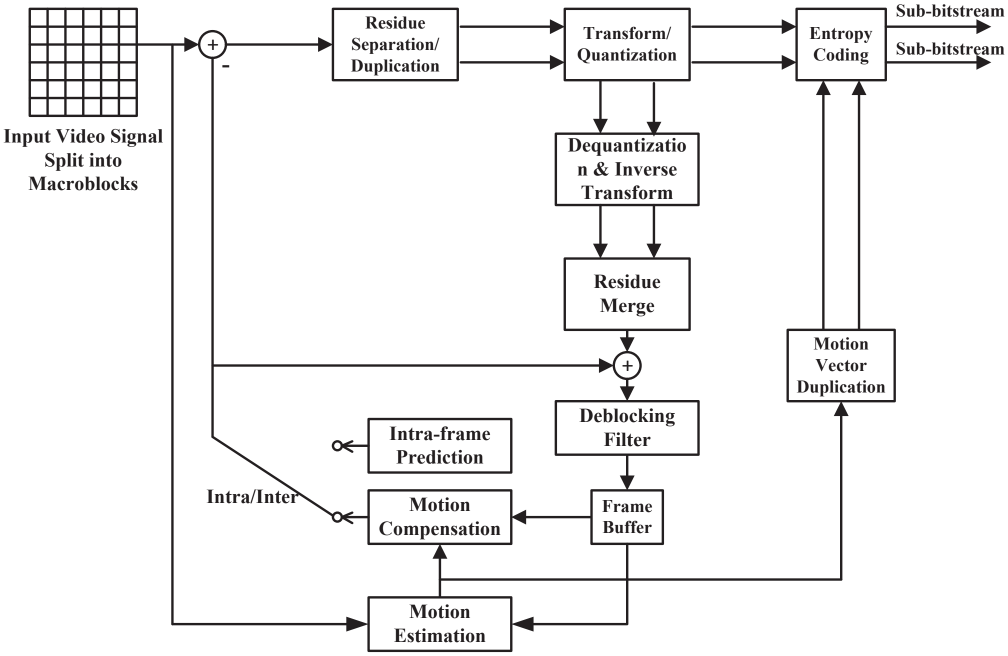


Fig. 6.2. MDC Encoder.

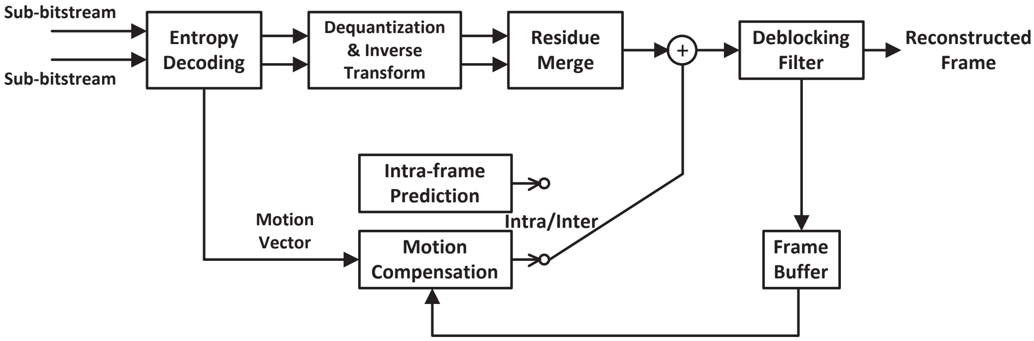


Fig. 6.3. MDC Decoder.

Odd_1, we use spatial concealment to conceal Odd_2 first and then use temporal concealment to conceal Even_1 and Even_2. When two descriptions are received, if the two descriptions are from the same spatial correlation such as Odd_1 and Odd_2, we use temporal concealment to conceal Even_1 and Even_2, otherwise we use spatial con-

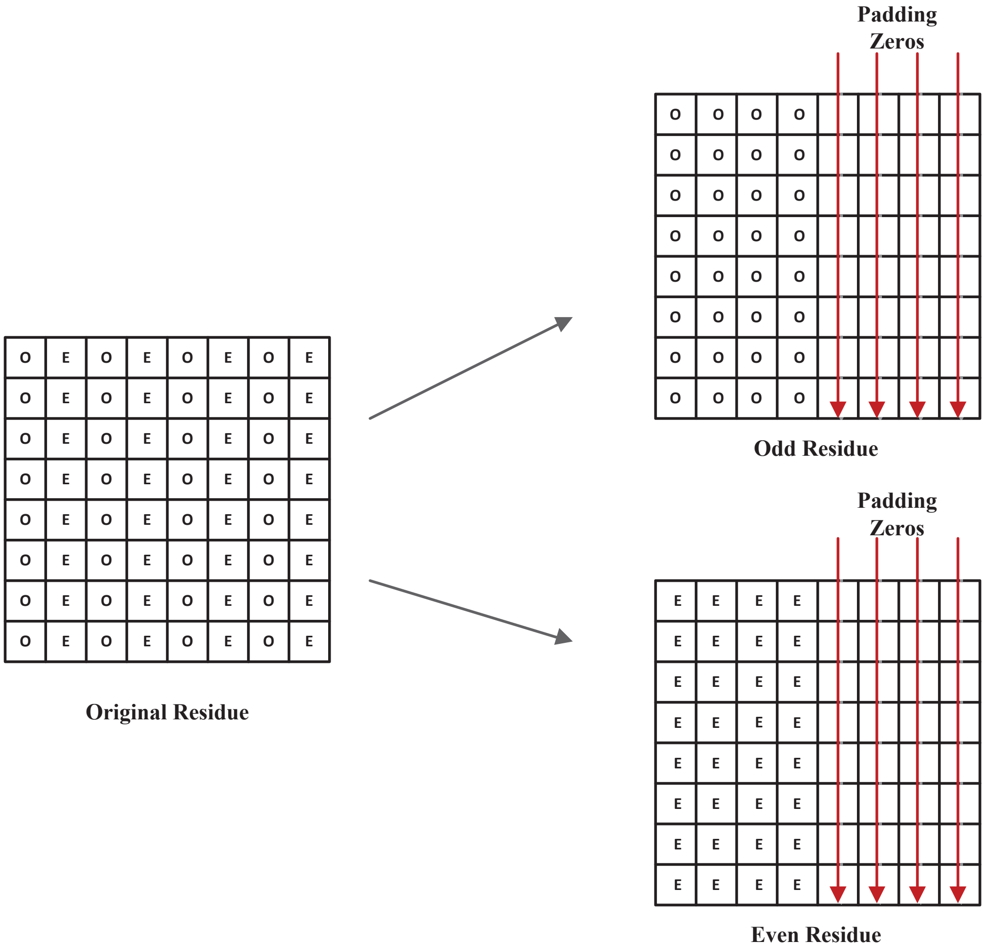


Fig. 6.4. MDC Spatial Separation.

cealment. When three descriptions are received, we use spatial concealment. And when the four descriptions are lost at the same time, we use the previously decoded reference frame for concealment which is named “Frame Repeat” in the table. In conclusion, spatial concealment is used as the default method and temporal concealment as the secondary method.

The error concealment schemes are the same as the schemes described in Chapter 3, but the effects are very different. This is because in our newly proposed MDC architecture, when one description of the two within the prediction loop is lost, for “skip” mode MB, the other description can provide all the information for decoding, therefore there is no influence for the decoding due to packet losses; for other inter prediction modes, as the other description within in the same prediction loop can provide motion vectors and all the control data to generate the prediction, the only thing we need to do is to estimate the lost part of the residues by spatial concealment. Note that when a skipped macroblock is signalled in the bitstream, no further data is sent for that macroblock. The decoder reconstructs the skipped macroblock using motion-compensated prediction [41]. For I frame, since it is duplicated into each of the two spatially correlated descriptions, this "key" frame is better protected.

The reason why we do not need to choose between spatial concealment and temporal concealment is that if temporal concealment is suitable for a lost macroblock, that means the macroblock is a low-motion MB, which in most of the time is encoded in "skip" mode.

For spatial concealment, we use a two-neighbor bilinear filter. For temporal concealment, we use a non-motion compensated method which copies the pixel value of the same position from the frame used for concealment.

6.3 Experimental Results

In this section, we evaluate the packet loss performance of the proposed method. The results are compared with H.264 and the MB-level adaptive method based on distortion mapping described in Chapter 5 which is among the best methods using the previous MDC partition architecture.

All the experiments are implemented by modifying the JVT JM software version 16.1. The testing sequences have original resolutions of CIF (352 × 288) at 30 frames/sec with 200-frame length. The coding structure is "IPPP...", with an I-frame refreshment every 30 frames. The quantization parameters for I frames and P frames are 18, 22, 26 and 30.

Three testing sequences are used in our experiments. The *"Mother"* sequence is a low-to-medium-motion sequence, *"Foreman"* is a medium-to-high-motion sequence and *"Soccer"* is a high-motion sequence. Experimental results are obtained by averaging 100 channel transmission simulations using Gilbert model. As noted in Section 3.3, when P_B is small, L_B is large; and vice versa [99]. The parameters for the Gilbert model in our experiments are listed in Table 6.2.

Table 6.2
Gilbert Model Parameters for Various Packet Loss Rates

Loss Rate	5%	10%	15%	20%
Burst Length	5	5	4	4

Figure 6.5 to 6.7 show packet loss performances of the *"Mother"*, *"Foreman"* and *"Soccer"* sequences respectively. Packet loss rates are from 5% to 20%.

Note that H.264 performs better than both the proposed method and the "MB-Distortion" method in terms of coding efficiency. Because, contrary to motion compression, error resilient video coding usually add redundancy to the bitstream to improve the error robustness, which inevitably decreases the coding efficiency. In the proposed method, the information used as reference for inter prediction is reduced. In the "MB-Distortion" method, the information used as reference for both intra prediction and inter prediction is reduced. Therefore, the prediction precision degrades and the residue increases which requires more bits to encode. However the coding efficiency of our proposed method is still better than the "MB-Distortion" method. This is because the method realizes the spatial partition on the residue and thus improves the prediction. When the prediction is more precise, it costs less bits to encode the transform coefficients.

When packet loss occurs, H.264 degrades more severely than the two proposed method and its packet loss performance is obviously worse when packet loss rate reaches 15%. This is because, four descriptions in our MDC model are sent to the decoder through different channels, which are unlikely to be lost at the same time. For each frame in a video sequence, as long as one description is received, the reconstructed frame at the decoder has an acceptable quality. However for H.264, when a frame is lost, the decoder can only use frame repeat to recover the lost frame.

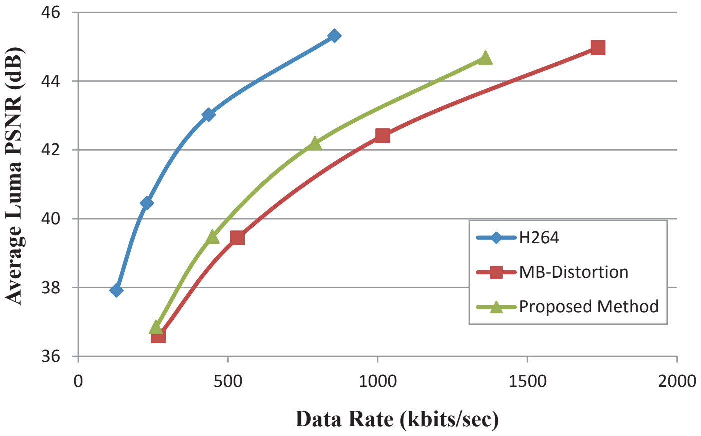
Figure 6.5 and Figure 6.6 show the results for the *"Mother"* and *"Foreman"* sequences respectively. The proposed method outperforms the "MB-Distortion" method in both coding efficiency and packet loss performance. This is because the proposed method realizes the spatial partition on the residue. And when the prediction is improved, the coding efficiency increases. In addition, the proposed method duplicates the I frame in each of the two descriptions within the same prediction loop so that this "key" frame is better protected than the "MB-Distortion" method, which improves the error robustness.

Figure 6.7 shows the results for the *"Soccer"* sequence. As is illustrated, the proposed method does not have an improvement over the "MB-Distortion" method. This is because as a high motion sequence, *"Soccer"* costs many bits to encode the MB partitions and motion vectors which are duplicated into each description within the same prediction loop in our method. Therefore, the cost of the additional bits compensates the bits saved on transform coefficients and therefore reduces the rate distortion performance.

Figure 6.8 and Figure 6.9 show the results for the *"Mother"* sequence when both the proposed method and the "MB-Distortion" method have identical packet loss positions.

Figure 6.8 shows the results when one of the two spatially correlated descriptions within the prediction loop is available for error concealment. In this situation, the proposed method is apparently sharper than the "MB-Distortion" method. This is because in our method, macroblocks encoded as "skip" mode can be perfectly recovered, and the macroblocks encoded as other inter prediction modes only need spatial concealment on the residues. For the "MB-Distortion" method, all the macroblocks require spatial concealment. The distortion of spatial concealment on pixels is larger than the distortion of spatial concealment on only residues.

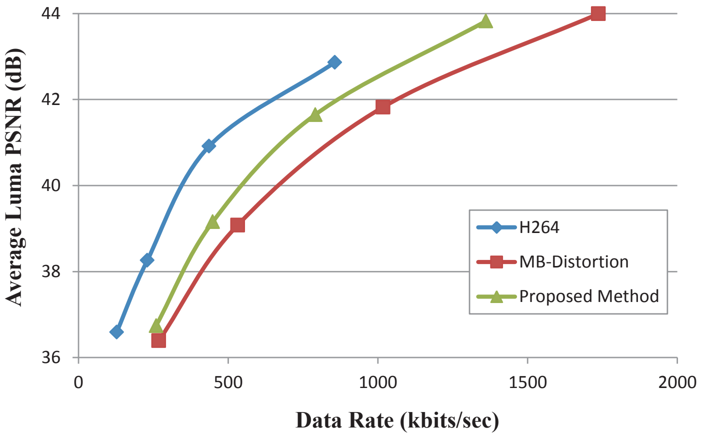
Figure 6.9 shows the results when several consecutive frames are lost before the one to be concealed. In this situation, the proposed method is more sensitive to error propagation than the "MB-Distortion" method. This is because, in our this method, if the spatially correlated description is received, the lost packet is always concealed using the spatially correlated description which is in the same prediction loop. However for the "MB-Distortion" method, both the spatially correlated description and the temporally correlated description are in independent prediction loops which can help the lost description "clear" the distortion due to its own error propagation.



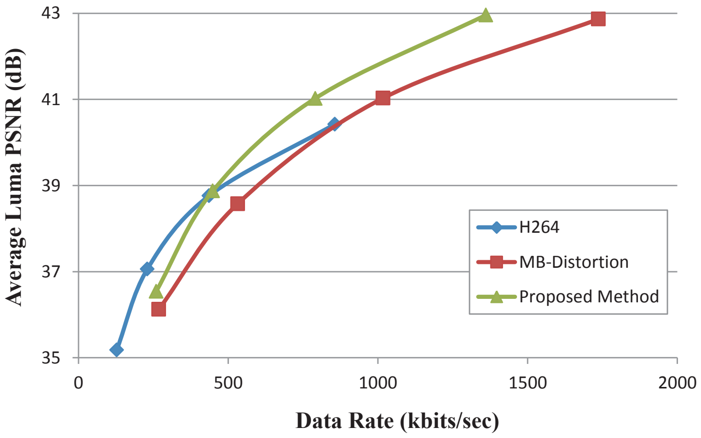
(a) Packet Loss = 0%.

6.4 Conclusion

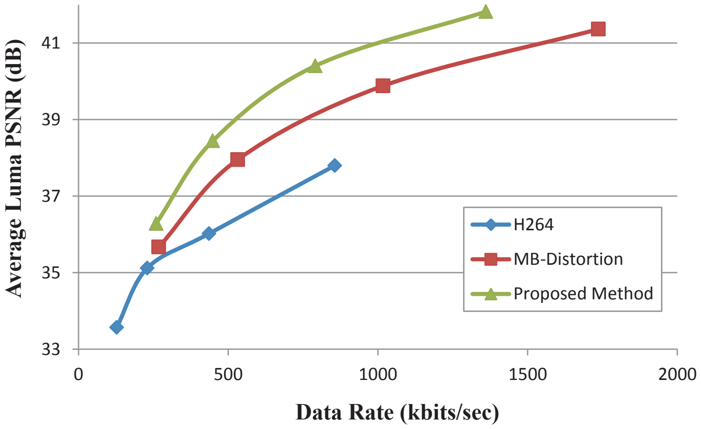
In this chapter, we described an advanced MDC architecture for improved coding efficiency and error robustness, in which temporal partition is done outside the prediction loop while spatial partition is done inside the prediction loop. Experimental results demonstrate the efficacy of the method. For objective quality, our method outperforms the MB-level adaptive method based on distortion mapping described in Chapter 5 which is among the best methods using the previous MDC partition archi-



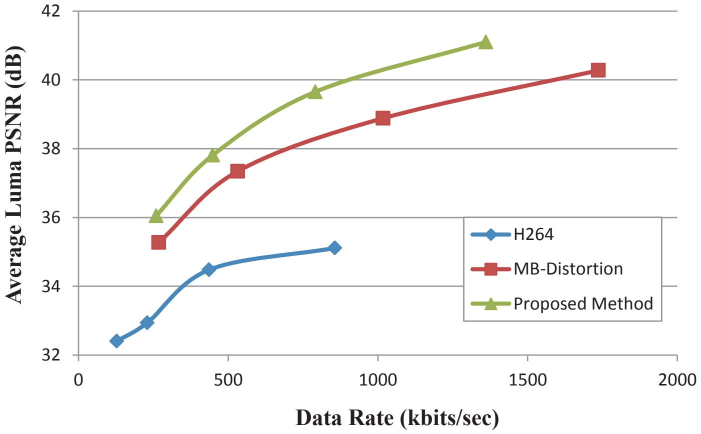
(b) Packet Loss = 5%.



(c) Packet Loss = 10%.

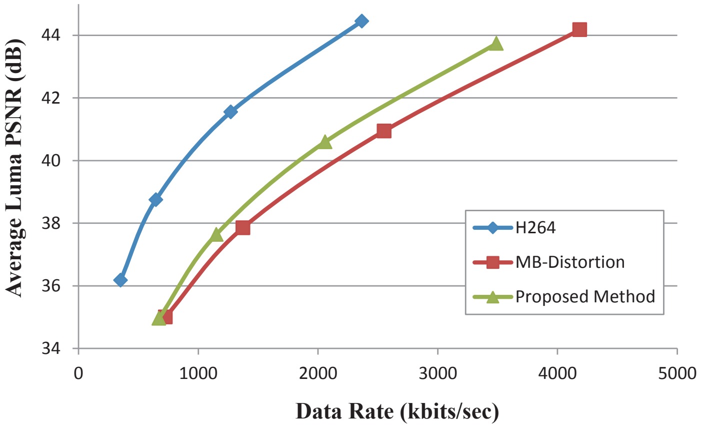


(d) Packet Loss = 15%.

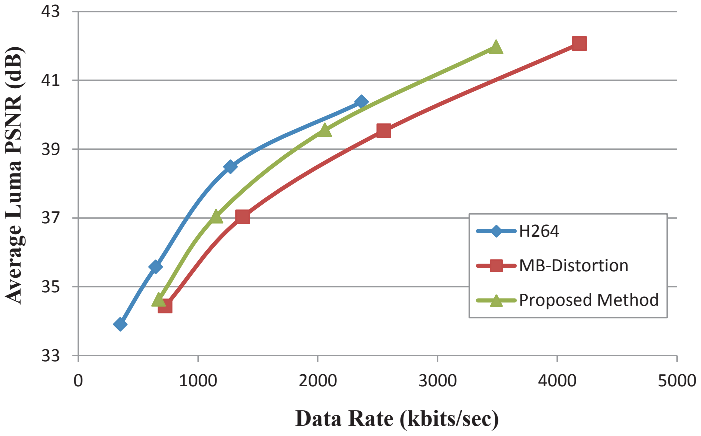


(e) Packet Loss = 20%.

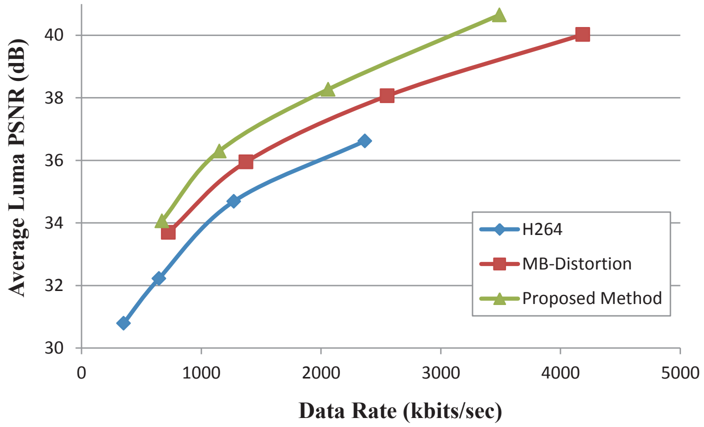
Fig. 6.5. Packet Loss Performances Comparison for the *"Mother"* Sequence (MB-Distortion denotes MB-level adaptive error concealment method for MDC based on distortion mapping).



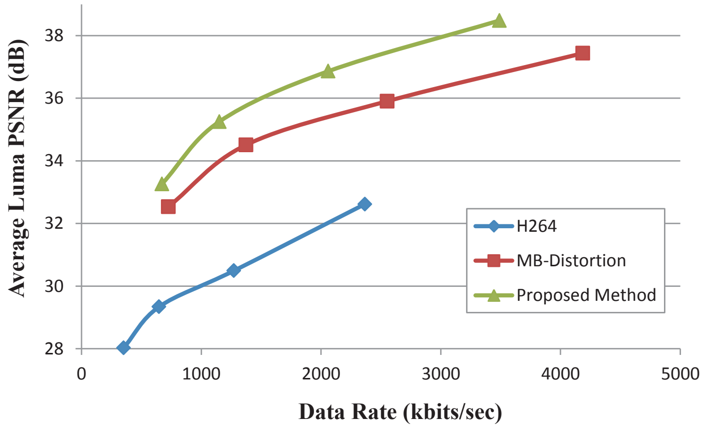
(a) Packet Loss = 0%.



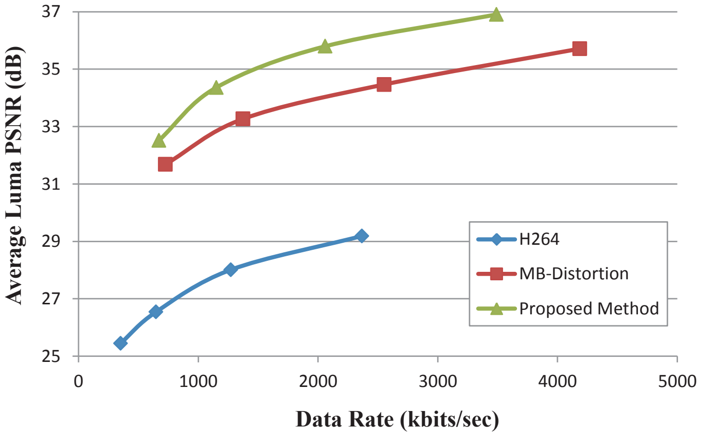
(b) Packet Loss = 5%.



(c) Packet Loss = 10%.

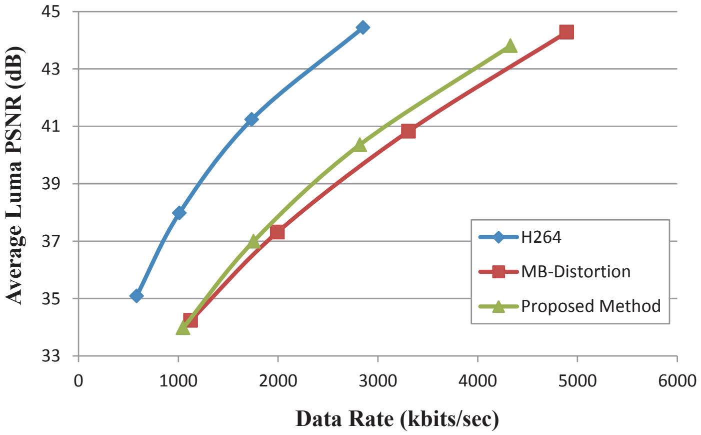


(d) Packet Loss = 15%.

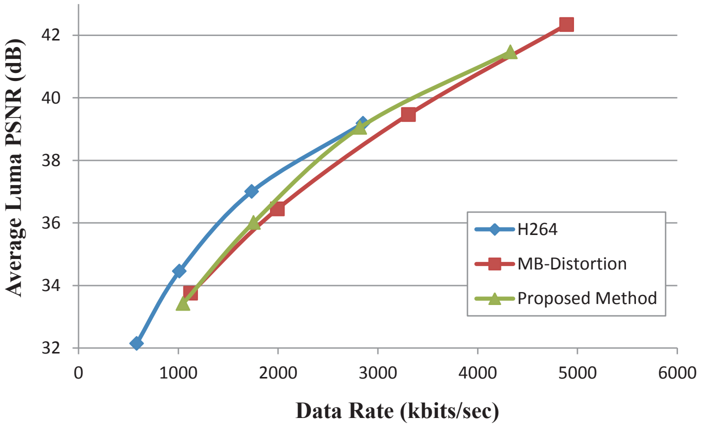


(e) Packet Loss = 20%.

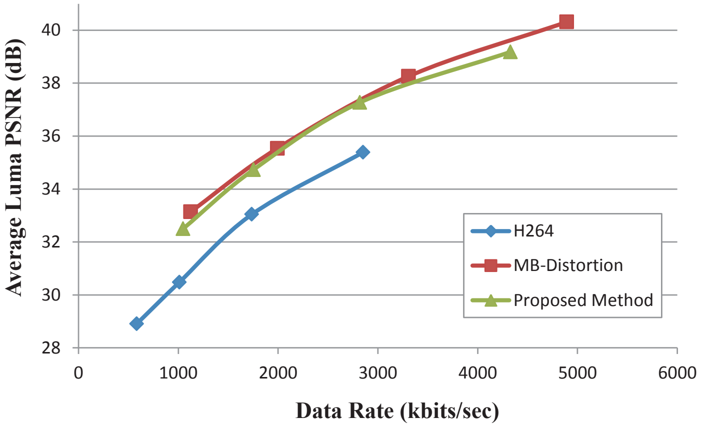
Fig. 6.6. Packet Loss Performances Comparison for the *"Foreman"* Sequence (MB-Distortion denotes MB-level adaptive error concealment method for MDC based on distortion mapping).



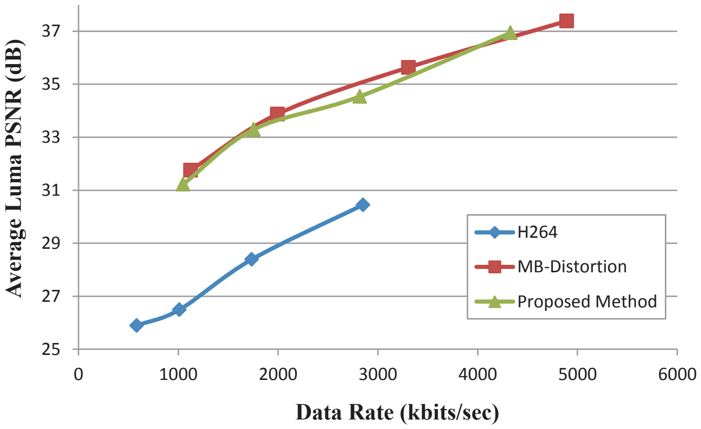
(a) Packet Loss = 0%.



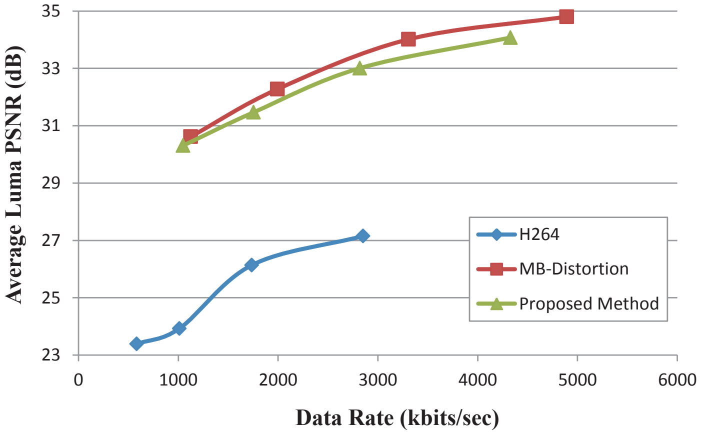
(b) Packet Loss = 5%.



(c) Packet Loss = 10%.

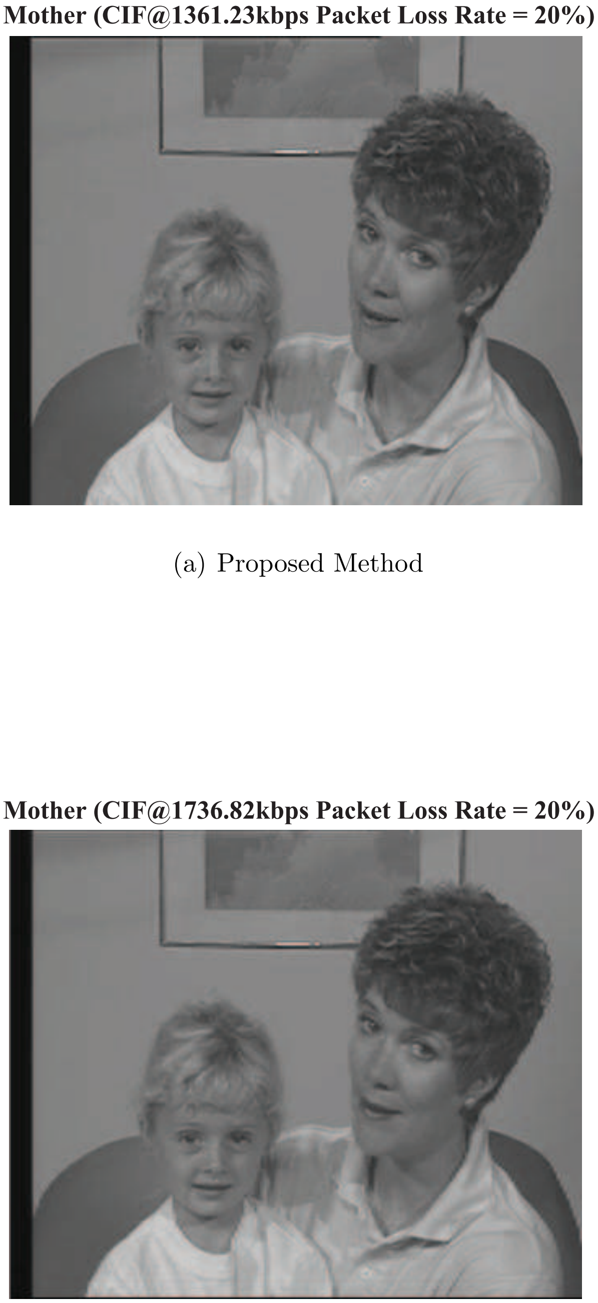


(d) Packet Loss = 15%.



(e) Packet Loss = 20%.

Fig. 6.7. Packet Loss Performances Comparison for the *"Soccer"* Sequence (MB-Distortion denotes MB-level adaptive error concealment method for MDC based on distortion mapping).



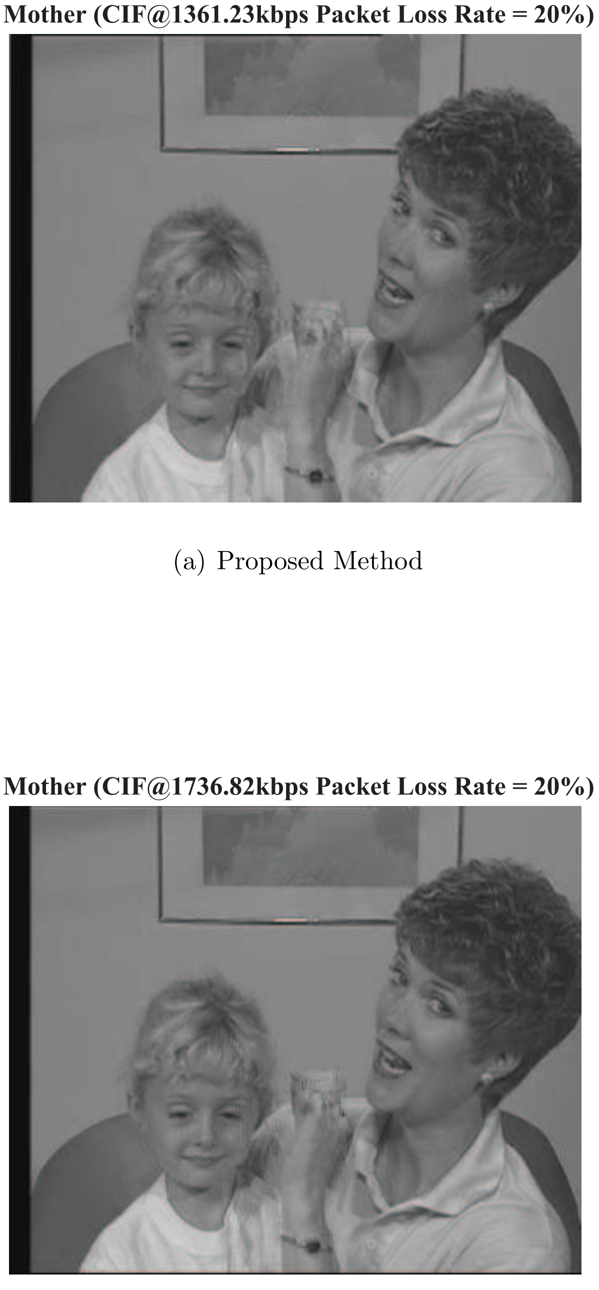
Mother (CIF@1361.23kbps Packet Loss Rate = 20%)

(a) Proposed Method

Mother (CIF@1736.82kbps Packet Loss Rate = 20%)

(b) "MB-Distortion" Method

Fig. 6.8. Packet Loss Performances Comparison for the *"Mother"* Sequence with Identical Packet Loss Positions in Terms of Sharpness.


Mother (CIF@1361.23kbps Packet Loss Rate = 20%)
Mother (CIF@1736.82kbps Packet Loss Rate = 20%)

(a) Proposed Method

(b) “MB-Distortion” Method

Fig. 6.9. Packet Loss Performances Comparison for the *“Mother”* Sequence with Identical Packet Loss Positions in Terms of Error Propagation Influence.

tecture. For visual quality, our proposed method is sharper than the “MB-Distortion” method, but is more sensitive to error propagation.

7. CONCLUSIONS

7.1 Summary

In this thesis, we investigated multiple description video coding with adaptive error concealment. The main contributions of this thesis are:

- Multiple Description Video Coding with Four Descriptions

  We proposed a new four-description MDC method based on a hybrid structure of temporal and spatial correlations with spatial concealment as the default method and temporal concealment as a secondary method. Experimental results show that both scalability performance and packet loss performance work well.

- An Adaptive Error Concealment for MDC Based on Error Tracking

  we invetigated an adaptable spatial-temporal error concealment method for multiple description video coding. This method takes into account the distortion due to error concealment and the distortion due to error propagation. Experimental results demonstrate that this method improves the non-adaptable method by adapting to any sequences.

- Macroblock-Level Adaptive Error Concealment Methods for MDC

  We described two adaptive spatial-temporal error concealment methods for MDC with four descriptions based on foreground-background mapping and distortion mapping respectively. Experimental results demonstrate that this method improves both the objective and subjective quality of the frame-level-concealment-based MDC.

- Advanced Multiple Description Video Coding for Better Coding Efficiency

  We investigated a new MDC partition architecture in which temporal separation is outside the prediction loop and spatial separation is inside the loop.

Experimental results show that this method improves the the objective quality but slightly degrades the subjective quality as a cost.

7.2 Future Work

Our methods can be improved and extended in the following ways:

- Since our work is about multiple description video coding with adaptive error concealment, we do not focus on error concealment. The performance should be improved if we combine error resilient video coding and error concealment. For simplicity we could use a bilinear filter for spatial concealment and non-motion compensated temporal concealment. In the future, one could use a more sophisticated spatial concealment such as a four-tap filter, a six-tap filter, bilateral filter, B-spline filter and O-moms filter. We could also develop a motion compensated temporal concealment by using frame interpolation or frame extension.

- We proposed two MDC partition architectures. One scheme performs better in terms of PSNR and the other performs better in terms of video quality. In the future work, we could develop an advanced adaptable error concealment method based on the MDC partition architecture that is illustrated in Chapter 6 which takes into account both objective distortion and subjective distortion. The cost function consists of two parts: PSNR which reflects the objective quality (OQ) and SSIM which reflects the subjective quality (SQ), expressed as $OQ + \lambda \times SQ$ where λ is a weighted factor. The error concealment method with the smallest cost is chosen for concealment.

- To further improve the performance of multiple description video coding, we could develop a backward channel aware MDC. Since we generate MDC based on the temporal and the spatial correlation, the four descriptions are highly correlated. When packet loss occurs, the decoder can recognize the lost positions for each description. Therefore, video quality can be estimated based on the types of the lost packets, the positions of the lost packets and whether spatial and/or temporal concealment are available for the lost packets (i.e., how much the lost packets can be concealed). The estimation of video quality by the decoder is sent back to the encoder for encoding improvements.

- In order to improve coding efficiency of multiple description video coding, we construct the spatial partition of the residue so that the two spatially correlated descriptions share a prediction loop. For error robustness, we protect the important syntax elements such as motion vectors, the macroblock header and I block transform coefficients by duplicating them into each description. This scheme is very similar to data partitioning in H.264/AVC. However, the bits saved using this architecture are compensated by these additional bits, and the benefits are not obvious in terms of coding efficiency. In the future, one could develop a scheme for better coding efficiency by splitting information such as motion vectors and I block transform coefficients into two descriptions. Since motion vectors or I block transform coefficients are highly self-correlated, the development of an advanced error concealment method to conceal these data can improve the coding efficiency while reducing slightly the error robustness.

7.3 Publications Resulting from the Thesis

Journal Articles:

- **Meilin Yang**, Mary L. Comer, and Edward J. Delp, "Temporal-Spatial Based Multiple Description Video Coding Methods with Adaptive Error Concealment," *IEEE Transactions on Circuits and Systems for Video Technology*, in preparation.

Conference Papers

- Neeraj Gadgil, **Meilin Yang**, Mary L. Comer, and Edward J. Delp, "Adaptive Error Concealment for Multiple Description Video Coding Using Motion Vector Analysis," *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, Florida, U.S.A., September, 2012.

- **Meilin Yang**, Mary L. Comer, and Edward J. Delp, "Macroblock-Level Adaptive Error Concealment Methods for MDC," *Proceedings of the Picture Coding Symposium (PCS)*, Krakow, Poland, May, 2012.

- **Meilin Yang**, Ye He, Fengqing Zhu, Marc Bosch, Mary L. Comer, and Edward J. Delp, "Video Coding: Death is Not Near," *Proceedings of the 53rd International Symposium ELMAR*, Zadar, Croatia, September 2011, pp. 85-88.

- **Meilin Yang**, Mary L. Comer, and Edward J. Delp, "An Adaptable Spatial-Temporal Error Concealment Method for Multiple Description Coding Based on Error Tracking," *Proceedings of IEEE International Conference on Image Processing (ICIP)*, Brussels, Belgium, September 2011, pp. 337-340.

- Rohan Gandhi, **Meilin Yang**, Dimitrios Koutsonikolas, Yu Charlie Hu, Mary Comer, Amr Mohamed, and Chih-Chun Wang, "The Impact of Inter-layer Network Coding on the Relative Performance of MRC/MDC WiFi Media Delivery," *Proceedings of the 21th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, Vancouver, Canada, June, 2011, pp. 27-32.

- **Meilin Yang**, Mary L. Comer, and Edward J. Delp, "A Four-Description MDC for High Loss-Rate Channels," *Proceedings of the Picture Coding Symposium (PCS)*, Nagoya, Japan, December 2010, pp. 418-421.

- Nitin Khanna, Fengqing Zhu, Marc Bosch, **Meilin Yang**, Mary L. Comer, and Edward J. Delp, "Information Theory Inspired Video Coding Methods: Truth Is Sometimes Better Than Fiction", *Proceedings of the Third Workshop on*

Information Theoretic Methods in Science and Engineering, Tampere, Finland, August, 2010.

LIST OF REFERENCES

LIST OF REFERENCES

[1] G. J. Sullivan and T. Wiegend, "Video compression: From concepts to the h.264/avc standard," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 18–31, January 2005.

[2] A. M. Tekalp, *Digital Video Processing.* Prentice Hall, 1995.

[3] T. Sikora, "Trends and perspectives in image and video coding," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 6–17, January 2005.

[4] A. C. Bovik, *Handbook of Image and Video Processing.* Academic Press, 2000.

[5] V. Bhaskaran and K. Konstantinides, *Image and Video Compression Standards: Algorithms and Architecture.* Boston, MA: Kluwer Academic Publishers, 1997.

[6] D. Tse and P. Viswanath, *Fundamentals of wireless communication.* Cambridge University Press, 2005.

[7] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the h.264/avc video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, July 2003.

[8] P. Yin, A. Vetro, B. Liu, and H. Sun, "Drift compensation for reduced spatial resolution transcoding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 11, pp. 1009–1020, November 2002.

[9] R. Schafer and T. Sikora, "Digital video coding standards and their role in video communications," *Proceedings of the IEEE*, vol. 83, no. 6, pp. 907–924, June 1995.

[10] A. Luthra, G. J. Sullivan, and T. Wiegand, "Introduction to the special issue on the h. 264/avc video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 557–559, 2003.

[11] G. J. Sullivan, P. Topiwala, and A. Luthra, "The h. 264/avc advanced video coding standard: Overview and introduction to the fidelity range extensions," *Proceedings of the SPIE Conference on Applications of Digital Image Processing XXVII*, vol. 5558, no. part 1, Denver, CA, 2004, pp. 454–474.

[12] G. J. Sullivan and T. Wiegand, "Video compression-from concepts to the h. 264/avc standard," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 18–31, 2005.

[13] K. Suhring, "H. 264/avc software coordination," http://iphome.hhi.de/suehring/tml/.

[14] I. Richardson, "H.264/mpeg-4 part 10 white paper," http://www.vcodex.fsnet.co.uk/resources.html.

[15] T. Wedi and H. G. Musmann, “Motion-and aliasing-compensated prediction for hybrid video coding,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 577–586, 2003.

[16] H. S. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky, “Low-complexity transform and quantization in h. 264/avc,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 598–603, 2003.

[17] D. Marpe, H. Schwarz, and T. Wiegand, “Context-based adaptive binary arithmetic coding in the h. 264/avc video compression standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 620–636, 2003.

[18] P. List, A. Joch, J. Lainema, G. Bjontegaard, and M. Karczewicz, “Adaptive deblocking filter,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 614–619, 2003.

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

[20] S. Wenger, “H. 264/avc over ip,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 645–656, 2003.

[21] T. Stockhammer, M. M. Hannuksela, and T. Wiegand, “H. 264/avc in wireless environments,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 657–673, 2003.

[22] S. Wenger, M. Hannuksela, T. Stockhammer, M. Westerlund, and D. Singer, “Rtp payload format for h. 264 video,” RFC 3984, Tech. Rep., 2005.

[23] “Narrow-band visual telephone systems and terminal equipment,” ITU-T Recommendation H.320, 1999.

[24] K. Brandenburg and G. Stoll, “The iso-mpeg-1 audio: A generic standard for coding of high-quality digital audio,” *Journal of the Audio Engineering Society*, vol. 42, no. 10, pp. 780–792, 1994.

[25] Y. Wang and Q. Zhu, “Error control and concealment for video communication: a review,” *Proceedings of the IEEE*, vol. 86, no. 5, pp. 974–997, May 1998.

[26] J. Yang, “A study of error-resilient interleaving with applications in the transmission of compressed images and video,” Ph.D. dissertation, Purdue University, 2004.

[27] P. Salama, “Error concealment in encoded images and video,” Ph.D. dissertation, Purdue University, 1999.

[28] S. Kumar, L. Xu, M. K. Mandal, and S. Panchanathan, “Error resiliency schemes in h. 264/avc standard,” *Journal of Visual Communication and Image Representation*, vol. 17, no. 2, pp. 425–450, 2006.

[29] T. Stockhammer, “Independent data partitions a and b,” JVT-C132, May 2002.

[30] G. Sullivan, “Seven steps toward a more robust codec design,” JVT-C117, May 2002.

[31] P. Lambert, W. D. Neve, Y. Dhondt, and R. V. de Walle, "Flexible macroblock ordering in h. 264/avc," *Journal of Visual Communication and Image Representation*, vol. 17, no. 2, pp. 358–375, 2006.

[32] W. Hantanong and S. Aramvith, "Analysis of macroblock-to-slice group mapping for h.264 video transmission over packet-based wireless fading channel," *Proceedings of the 48th Midwest Symposium on Circuits and Systems*, vol. 2, Cincinnati, OH, August 2005, pp. 1541–1544.

[33] J. Panyavaraporn, R. Cajote, and S. Aramvith, "Performance analysis of flexible macroblock ordering using bit-count and distortion measure for h.264 wireless video transmission," *Proceedings of the International Workshop on Smart Info-Media Systems in Bangkok*, Bangkok, Thailand, 2007.

[34] R. D. Cajote, S. Aramvith, R. C. L. Guevara, and Y. Miyanaga, "Fmo slice group maps using spatial and temporal indicators for h.264 wireless video transmission," *Proceedings of the IEEE International Symposium on Circuits and Systems*, Seattle, WA, 18-21 2008, pp. 3566–3569.

[35] J. Li and K. N. Ngan, "An error sensitivity-based redundant macroblock strategy for robust wireless video transmission," *Proceedings of the 2005 International Conference on Wireless Networks*, vol. 2, Las Vegas, NV, June 2005, pp. 1118–1123.

[36] Y. Dhondt, P. Lambert, and R. V. de Walle, "A flexible macroblock scheme for unequal error protection," *Proceedings of the 2006 IEEE International Conference on Image*, Atlanta, GA, October 2006, pp. 829–832.

[37] S. Im and A. J. Pearmain, "Unequal error protection with the h.264 flexible macroblock ordering," *Proceedings of SPIE Conference on Visual Communications and Image Processing*, Beijing, China, 2005.

[38] N. Thomos, S. Argyropoulos, N. V. Boulgouris, and M. G. Strintzis, "Error-resilient transmission of h.264/avc streams using flexible macroblock ordering," *Proceedings of the 2nd European Workshop on the Integration of Knowledge, Semantics and Digital Media Technology*, London, UK, November 2005, pp. 183–189.

[39] S. Wenger and M. Horowitz, "Fmo: Flexible macroblock ordering," *JVT-C089*, May 2002.

[40] M. Karczewicz and R. Kurceren, "The sp-and si-frames design for h. 264/avc," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 637–644, 2003.

[41] I. E. G. Richardson, *H.264 and MPEG-4 video compression.* Wiley Online Library, 2003, vol. 20.

[42] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the h.264/avc standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, September 2007.

[43] R. Zhang and M. L. Comer, "Subband motion compensation for spatially scalable video coding," *Proceedings of SPIE on Visual, Communication, and Image Processing*, San Jose, CA, 2007.

[44] ——, “Improved motion compensation in the enhancement layer for spatially scalable video coding,” *Proceedings of the IEEE International Conference on Image Processing*, vol. 2, San Antonio, TX, 2007, pp. 301–304.

[45] ——, “Adaptive enhancement layer motion compensation for video spatial scalability,” *Proceedings of the IEEE International Conference on Image Processing*, San Diego, CA, 2008, pp. 1212–1215.

[46] ——, “Efficient inter-layer motion compensation for spatially scalable video coding,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 10, pp. 1325–1334, October 2008.

[47] ——, “Rate distortion analysis of subband motion compensation for video spatial scalability,” *Proceedings of the 2009 Picture Coding Symposium*, pp. 1–4, May 2009.

[48] ——, “Rate distortion analysis for spatially scalable video coding,” *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 2947–2957, 2010.

[49] ——, “Rate distortion performance of pyramid and subband motion compensation based on quantization theory,” *IEEE Transactions on Circuits and Systems for Video Technology*, no. 99, pp. 1876–1881, 2010.

[50] G. Cook, J. Prades-Nebot, Y. Liu, and E. Delp, “Rate-distortion analysis of motion-compensated rate scalable video,” *IEEE Transactions on Image Processing*, vol. 15, no. 8, pp. 2170–2190, August 2006.

[51] J. Prades-Nebot, G. W. Cook, and E. J. Delp, “An analysis of the efficiency of different snr-scalable strategies for video coders,” *IEEE Transactions on Image Processing*, vol. 15, no. 4, pp. 848–864, April 2006.

[52] Y. Liu, P. Salama, Z. Li, and E. Delp, “An enhancement of leaky prediction layered video coding,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 11, pp. 1317–1331, November 2005.

[53] J. R. Ohm, “Advances in scalable video coding,” *Proceedings of the IEEE*, vol. 93, no. 1, pp. 42–56, January 2005.

[54] M. Comer, “Conditional replacement for improved coding efficiency in fine-grain scalable video coding,” *Proceedings of the International Conference on Image Processing*, vol. 2, Rochester, NY, 2002, pp. II–57 – II–60.

[55] K. Shen and E. J. Delp, “Wavelet based rate scalable video compression,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 1, pp. 109 – 122, February 1999.

[56] L. A. Overturf, M. L. Comer, and E. J. Delp, “Color image coding using morphological pyramid decomposition,” *IEEE Transactions on Image Processing*, vol. 4, no. 2, pp. 177–185, Feburary 1995.

[57] K. Hua, I. Pollak, and M. Comer, “Optimal tilings for image and video compression,” *Proceedings of the Asilomar Conference on Signals, Systems and Computers*, Pacific Groove, CA, October 2006, pp. 391 –395.

[58] Y. Wang, J. Ostermann, and Y. Zhang, *Video processing and communications.* Prentice Hall, 2002, vol. 5.

[59] Y. Wang, A. R. Reibman, and S. Lin, “Multiple description coding for video delivery,” *Proceedings of the IEEE*, vol. 93, no. 1, pp. 57 –70, January 2005.

[60] V. Goyal, “Multiple description coding: compression meets the network,” *IEEE Transactions on Signal Processing*, vol. 18, no. 5, pp. 74 –93, September 2001.

[61] V. Vaishampayan, “Design of multiple description scalar quantizers,” *IEEE Transactions on Information Theory*, vol. 39, no. 3, pp. 821 –834, May 1993.

[62] V. A. Vaishampayan and J. Domaszewicz, “Design of entropy-constrained multiple-description scalar quantizers,” *IEEE Transactions on Information Theory*, vol. 40, no. 1, pp. 245–250, 1994.

[63] V. A. Vaishampayan and J. C. Batllo, “Multiple description transform codes with an application to packetized speech,” *Proceedings of the International Symposium on Information Theory*, Trondheim, Norway, July 1994, p. 458.

[64] V. A. Vaishampayan, “Application of multiple description codes to image and video transmission over lossy networks,” *Procceedings of the 7th International Workshop Packet Video*, 1996, pp. 55–60.

[65] J. C. Batllo and V. A. Vaishampayan, “Asymptotic performance of multiple description transform codes,” *IEEE Transactions on Information Theory*, vol. 43, no. 2, pp. 703–707, 1997.

[66] Y. Wang, M. T. Orchard, and A. R. Reibman, “Multiple description image coding for noisy channels by pairing transform coefficients,” *Proceedings of the IEEE First Workshop on Multimedia Signal Processing*, Princeton, NJ, 1997, pp. 419–424.

[67] M. T. Orchard, Y. Wang, V. Vaishampayan, and A. R. Reibman, “Redundancy rate-distortion analysis of multiple description coding using pairwise correlating transforms,” *Proceedings of the International Conference on Image Processing*, vol. 1, Washington, DC, 1997, pp. 608–611.

[68] Y. Wang, M. Orchard, V. Vaishampayan, and A. Reibman, “Multiple description coding using pairwise correlating transforms,” *IEEE Transactions on Image Processing*, vol. 10, no. 3, pp. 351 –366, March 2001.

[69] A. Reibman, H. Jafarkhani, Y. Wang, M. Orchard, and R. Puri, “Multiple description coding for video using motion compensated prediction,” *Proceedings of the 1999 International Conference on Image Processing*, vol. 3, Kobe, Japan, October 1999, pp. 837–841.

[70] S. S. Hemami, “Reconstruction-optimized lapped orthogonal transforms for robust image transmission,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 2, pp. 168–181, 1996.

[71] H. S. Malvar and D. H. Staelin, “The lot: Transform coding without blocking effects,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 4, pp. 553–559, 1989.

[72] T. Tillo and G. Olmo, "Data-dependent pre-and postprocessing multiple description coding of images," *IEEE Transactions on Image Processing*, vol. 16, no. 5, pp. 1269–1280, 2007.

[73] S. Shirani, M. Gallant, and F. Kossentini, "Multiple description image coding using pre-and post-processing," *Proceedings of the International Conference on Information Technology: Coding and Computing*, Las Vegas, NV, 2001, pp. 35–39.

[74] M. Gallant, S. Shirani, and F. Kossentini, "Standard-compliant multiple description video coding," *Proceedings of the International Conference on Image Processing*, vol. 1, Thessaloniki, Greece, 2001, pp. 946–949.

[75] N. Franchi, M. Fumagalli, and R. Lancini, "Flexible redundancy insertion in a polyphase down sampling multiple description image coding," *Proceedings of the IEEE International Conference on Multimedia and Expo*, vol. 2, Lausanne, Switzerland, August 2002, pp. 605 – 608.

[76] Z. Wei, C. Cai, and K.-K. Ma, "A novel h.264-based multiple description video coding via polyphase transform and partial prediction," *Proceedings of the International Symposium on Intelligent Signal Processing and Communications*, vol. 1, Yonago, Japan, December 2006, pp. 151 –154.

[77] N. Franchi, M. Fumagalli, R. Lancini, and S. Tubaro, "Multiple description video coding for scalable and robust transmission over ip," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 3, pp. 321 – 334, March 2005.

[78] S. Shirani, "Content-based multiple description image coding," *IEEE Transactions on Multimedia*, vol. 8, no. 2, pp. 411–419, 2006.

[79] D. Wang, N. Canagarajah, D. Redmill, and D. Bull, "Multiple description video coding based on zero padding," *Proceedings of the 2004 International Symposium on Circuits and Systems*, vol. 2, Vancouver, BC , Canada, 2004, pp. 205–208.

[80] J. G. Apostolopoulos, "Error-resilient video compression through the use of multiple states," *Proceedings of the 2000 International Conference on Image Processing*, vol. 3, Vancouver, BC , Canada, 2000, pp. 352 –355.

[81] G. Zhang and R. L. Stevenson, "Efficient error recovery for multiple description video coding," *Proceedings of the 2004 International Conference on Image Processing*, vol. 2, Singapore, 2004, pp. 829–832.

[82] I. V. Bajic and J. W. Woods, "Domain-based multiple description coding of images and video," *IEEE Transactions on Image Processing*, vol. 12, no. 10, pp. 1211–1225, 2003.

[83] M. V. der Schaar and D. S. Turaga, "Multiple description scalable coding using wavelet-based motion compensated temporal filtering," *Proceedings of the 2003 International Conference on Image Processing*, vol. 3, Barcelona, Catalonia, Spain, 2003, pp. 489–492.

[84] E. Akyol, A. M. Tekalp, and M. R. Civanlar, “Scalable multiple description video coding with flexible number of descriptions,” *Proceedings of the 2005 International Conference on Image Processing*, vol. 3, Genoa, Italy, 2005, pp. 712–715.

[85] E. Akyol, A. Tekalp, and M. Civanlar, “A flexible multiple description coding framework for adaptive peer-to-peer video streaming,” *IEEE Journal of Selected Topics in Signal Processing,*, vol. 1, no. 2, pp. 231 –245, August 2007.

[86] N. V. Boulgouris, K. E. Zachariadis, A. N. Leontaris, and M. G. Strintzis, “Drift-free multiple description coding of video,” *Proceedings of the IEEE Fourth Workshop on Multimedia Signal Processing*, Cannes, France, 2001.

[87] A. Reibman, H. Jafarkhani, Y. Wang, and M. Orchard, “Multiple description video using rate-distortion splitting,” *Proceedings of the 2001 International Conference on Image Processing*, vol. 1, Thessaloniki , Greece, October 2001, pp. 978–981.

[88] C. Kim and S. Lee, “Multiple description coding of motion fields for robust video transmission,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 9, pp. 999 –1010, sep 2001.

[89] Y. Wang and S. Lin, “Error-resilient video coding using multiple description motion compensation,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 6, pp. 438–452, June 2002.

[90] C. Zhu and M. Liu, “Multiple description video coding based on hierarchical b pictures,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 4, pp. 511–521, April 2009.

[91] R. Puri and K. Ramchandran, “Multiple description source coding using forward error correction codes,” *Conference Record of the Thirty-Third Asilomar Conference on Signals, Systems, and Computers*, vol. 1, Pacific Grove, CA, 1999, pp. 342–346.

[92] R. Bernardini, M. Durigon, R. Rinaldo, L. Celetto, and A. Vitali, “Polyphase spatial subsampling multiple description coding of video streams with h264,” *Proceedings of the International Conference on Image Processing*, vol. 5, Singapore, October 2004, pp. 3213–3216.

[93] J. R. Ohm, “Three-dimensional subband coding with motion compensation,” *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 559–571, 1994.

[94] C.-W. Hsiao and W.-J. Tsai, “Hybrid multiple description coding based on h.264,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 1, pp. 76 –87, January 2010.

[95] M. Yang, M. Comer, and E. J. Delp, “A four-description mdc for high loss-rate channels,” *Proceedings of the 2010 Picture Coding Symposium*, Nagoya, Japan, December 2010, pp. 418–421.

[96] N. Khanna, F. Zhu, M. Bosch, M. Yang, M. Comer, and E. J. Delp, “Information theory inspired video coding methods: Truth is sometimes better than fiction,” *Proceedings of the Third Workshop on Information Theoretic Methods in Science and Engineering*, Tampere, Finland, August 2010.

[97] M. Yang, Y. He, F. Zhu, M. Bosch, M. Comer, and E. J. Delp, “Video coding: Death is not near,” *Proceedings of the 53rd International Symposium ELMAR*, Zadar, Croatia, September 2011, pp. 85–88.

[98] R. Zhang, “Efficient inter-layer motion compensation and error resilience for spatially scalable video coding,” Ph.D. dissertation, Purdue University, 2009.

[99] U. Horn, K. Stuhlmller, M. Link, and B. Girod, “Robust internet video transmission based on scalable coding and unequal error protection,” *Image Communication*, vol. 15, pp. 77–94, September 1999.

[100] M. Yang, M. L. Comer, and E. J. Delp, “An adaptable spatial-temporal error concealment method for multiple description coding based on error tracking,” *Proceedings of the IEEE International Conference on Image Processing*, Brussels, Belgium, September 2011, pp. 337–340.

[101] ——, “Macroblock-level adaptive error concealment methods for mdc,” *Proceedings of the 2012 Picture Coding Symposium*, Krakow, Poland, May 2012.

VITA

VITA

Meilin Yang was born in Harbin, Heilongjiang Province, China. She received her Bachelor of Science degree in Electronics and Information Engineering from Harbin Institute of Technology, Harbin, Heilongjiang Province, China in 2008.

Meilin joined the Ph.D. program at Purdue University, West Lafayette, Indiana in 2008. Since 2009, she has served as Research Assistant at the Video and Image Processing Laboratory (VIPER). She is co-supervised by Professor Mary Comer who is Associate Professor of Electrical and Computer Engineering and Professor Edward J. Delp who is the Charles William Harrison Distinguished Professor of Electrical and Computer Engineering and Professor of Biomedical Engineering. While in the graduate program, Meilin has worked on projects sponsored by grants from Cisco System Incorporated. During the summer of 2011, she was a Student Intern at the Qualcomm Incorporated, San Diego, CA. Her research interests include video compression, video transmission, image/video processing, and computer vision.

Meilin is a student member of the IEEE and the IEEE Signal Processing Society.