Error Resilient Methods in Scalable Video Coding (SVC)

Amir Naghdinezhad

Department of Electrical & Computer Engineering
McGill University
Montreal, Canada

December 2013

A thesis submitted to McGill University in partial fulfilment of the requirements for the degree of Doctor of Philosophy.

2013/12/02

Abstract

With the rapid development of multimedia technology, video transmission over unreliable channels like Internet and wireless networks, is widely used. Channel errors can result in a mismatch between the encoder and the decoder, and because of the predictive structures used in video coding, the errors will propagate both temporally and spatially. Consequently, the quality of the received video at the decoder may degrade significantly. In order to improve the quality of the received video, several error resilient methods have been proposed. Furthermore, in addition to compression efficiency and error robustness, flexibility has become a new multimedia requirement in advanced multimedia applications. In these applications such as video conferencing and video streaming, compressed video is transmitted over heterogeneous networks with a broad range of clients with different requirements and capabilities in terms of power, bandwidth and display resolution, simultaneously accessing the same coded video. The scalable video coding concept was proposed to address the flexibility issue by generating a single bit stream that meets the requirement of these users.

This dissertation is concerned with novel contributions in the area of error resilience for scalable extension of H.264/AVC. The first part of the dissertation focuses on modifying the conventional prediction structure in order to reduce the propagation of error to succeeding frames. We propose two new prediction structures that can be used in temporal and spatial scalability of SVC. The proposed techniques improve the previous methods by efficiently exploiting the Intra macroblocks (MBs) in the reference frames and exponential decay of error propagation caused by the introduced leaky prediction.

In order to satisfy both coding efficiency and error resilience in error prone channels, we combine error resilience mode decision technique with the proposed prediction structures. The end-to-end distortion of the proposed prediction structure is estimated and used instead of the source coding distortion in the rate distortion optimization.

Furthermore, accurately analysing the utility of each video packet in unequal error protection techniques is a critical and usually very complex process. We present an accurate low complexity utility estimation technique. This technique estimates the utility of each network abstraction layer (NAL) by considering the error propagation to future frames. Also, a low delay version of this technique, which can be used in delay constrained applications, is presented.

Sommaire

La révolution technologique de l'information et des communications a donné lieu à un élargissement du marché des applications multimédias. Sur des canaux non fiables comme Internet et les réseaux sans fil, la présence des erreurs de transmission est considérée comme l'une des principales causes de la dégradation de la qualité vidéo au niveau du récepteur. Et en raison des structures de prédiction utilisées dans le codage vidéo, ces erreurs ont tendance à se propager à la fois temporellement et spatialement. Par conséquent, la qualité de la vidéo reçue risque de se dégrader d'une façon considérable. Afin de minimiser ce risque, des outils qui permettent de renforcer la robustesse contre les erreurs ont été proposés. En plus de la résistance aux erreurs, la flexibilité est devenue une nouvelle exigence dans des applications multimédias comme la vidéo conférence et la vidéo en streaming. En effet, la vidéo compressée est transmise sur des réseaux hétérogènes avec un large éventail de clients ayant des besoins différents et des capacités différentes en termes de puissance, de résolution vidéo et de bande passante, d'où la nécessite d'une solution pour l'accès simultané à la même vidéo codée. La scalabilité est venue répondre aux exigences de tous ces utilisateurs.

Cette thèse, élaborée dans le cadre du développement de la version scalable de la norme H.264/AVC (aussi connue sous le nom de SVC), présente des idées innovantes dans le domaine de la résilience aux erreurs. La première partie de la thèse expose deux nouvelles structures de prédiction qui aident à renforcer la résistance aux erreurs. Les structures proposées peuvent être utilisées dans la scalabilité temporelle et spatiale et visent essentiellement à améliorer les méthodes antérieures en exploitant de manière plus efficace les MBs "Intra" dans les images de référence et en profitant de la prédiction "Leaky" qui permet de réduire de façon exponentielle la propagation des erreurs de transmission.

Afin de satisfaire à la fois l'efficacité du codage et la résilience aux erreurs, nous avons combiné les techniques proposées avec les modules de décision. En plus, une estimation de la distorsion de bout en bout a été utilisée dans le calcul du coût des différents modes. En outre, analyser avec précision l'importance de chaque paquet de données vidéo dans de telles structures est un processus critique et généralement très complexe. Nous avons proposé une méthode simple et fiable pour cette estimation. Cette méthode consiste à évaluer l'importance de chaque couche d'abstraction réseau (NAL) en considérant la propagation des erreurs dans les images futures. En plus, une version avec un faible délai de réponse a

été présentée.

Acknowledgements

I am in debt to my supervisor, Professor Fabrice Labeau, for helping me to define and refine my ideas on this dissertation. This work would have not been possible without his support, dedication, inspiration, and helpful ideas. I extremely value and appreciate his trust and the freedom he allowed me during my doctoral studies.

My special thanks go to co-supervisor, Professor Leszek Szczecinski for providing constructive feedbacks on my ideas and results. I would like to thank my PhD committee, Professor Peter Kabal, and Professor Benoit Boulet for their valuable feedback on my research.

I gratefully acknowledge the financial support received from the Natural Sciences and Engineering Research Council (NSERC) and industrial and government partners, through the Healthcare Support through Information Technology Enhancements (hSITE) Strategic Research Network.

Many individuals had a constructive influence on my work. I would like to specially thank Dr. Ramdas Satyan and Dr. Sunday Nyamweno for their help to refine and revise my ideas in this thesis. I would like to thank Dr. Michael Horowitz for his positive influence and insight in my research ideas in the Video Coding area. I am grateful for endless support and encouragement from my friends and colleagues at McGill: Mehdi, Mohsen, Mahdy, Siavash, Leila and Sina to mention a few. I would also like to thank Hsan Guermezi for the French translation of this dissertation's abstract.

On a personal note, I like to thank my parents, my brother, and my sister. I am grateful for their support and encouragements. Last, but never least, to my beloved partner Golnaz: Thank you for your patience, support and for always believing in me.

Contents

List of Figures

List of Tables

List of Acronyms

4CIF	4 x Common Intermediate Format
ARQ	Automatic retransmission on request
BWDE	Block weighted distortion estimate
CIF	Common Intermediate Format
CGS	Course Grain Scalability
CRC	Cyclic Redundancy Check
DPB	Decoding Picture Buffer
DCT	Discrete Cosine Transform
ER-RDO	Error Robust Rate Distortion Optimization
FGS	Fine Grain Scalability
FMO	Flexible macroblock ordering
FEC	Forward Error Correction
fps	Frames Per Second
GSCP	Generalized Source Channel Prediction
GA	Genetic Algorithm
GOP	Group of Pictures
IGSCP	Improved Generalized Source Channel Prediction
JSVM	Joint Scalable Video Model
kbps	Kilo Bits per Second
LW-EZEP	Layer-Weighted Expected Zone of Error Propagation
LARDO	Loss Aware Rate Distortion Optimization
LDPC	Low Density Parity Check
MB	MacroBlock
MSE	Mean Square Error

MGS	Medium Grain Scalability
MCP	Motion Compensated Prediction
MC	Motion Copy
ME	Motion Estimation
MV	Motion Vector
MPEG	Moving Picture Experts Group
MHMCP	Multi-Hypothesis Motion Compensated Prediction
MC	Multiple description coding
NAL	Network Abstraction Layer
PLR	Packet Loss Rate
PSO	Particle Swarm Optimization
PSNR	Peak Signal to Noise Ratio
PC	Picture Copy
PPS	Picture Parameters Set
PSR	Probability of Successfully Receiving
QoS	Quality of Service
QP	Quantization Parameter
QCIF	Quarter Common Intermediate Format
ROPE	Recursive Optimal per-Pixel Estimate
RS	Reed Solomon
RFM	Reference Frame Modification
SVC	Scalable Video Coding
SPS	Sequence Parameter Set
SAD	Sum of Absolute Differences
SSD	Sum of Square Differences
SEI	Supplemental Enhancement Information
TEU	Total Expected Utility
TSD	Total Sequence Distortion
UEP	Unequal Error Protection
VCEG	Video Coding Experts Group

List of Symbols

α	Leaky factor
β	The weight of $\hat{K}_n^i$
λ_{mode}	Lagrangian multiplier for the mode decision optimization
λ_{motion}	Lagrangian multiplier for the motion estimation optimization
B_{C_i}	The number of channel bits for the i^{th} NAL unit
B_{S_i}	The number of source bits for the i^{th} NAL unit
D	Total distortion
$d(n,i)$	The end-to-end distortion of the i^{th} pixel in the n^{th} frame
$D(n,m)$	The end-to-end distortion of the m^{th} block in the n^{th} frame
$d_{\text{ec\_org}}(n,i)$	The original frame error concealment distortion of the i^{th} pixel in the n^{th} frame
$D_{\text{ec\_org}}(n,m)$	The original frame error concealment distortion of the m^{th} block in the n^{th} frame
$d_{\text{ec\_rec}}(n,i)$	The reconstructed frame error concealment distortion of the i^{th} pixel in the n^{th} frame
$D_{\text{ec\_rec}}(n,m)$	The reconstructed frame error concealment distortion of the m^{th} block in the n^{th} frame
$d_{\text{ec}}(n,i)$	The error concealment distortion of the i^{th} pixel in the n^{th} frame
$D_{\text{ec}}(n,m)$	The error concealment distortion of the m^{th} block in the n^{th} frame
$d_{\text{ep}}(n,i)$	The error propagation distortion of the i^{th} pixel in the n^{th} frame
$d'_{\text{ep}}(n,i)$	The error propagation distortion of pixel i in the modified reconstructed frame n at the encoder
$D_{\text{ep}}(n,m)$	The error propagation distortion of the m^{th} block in the n^{th} frame
$D_{\text{src}}(n)$	The source coding distortion for frame n

$d_{\text{src}}(n,i)$	The source coding distortion of the i^{th} pixel in the n^{th} frame
$D_{\text{src}}(n,m)$	The source coding distortion of the m^{th} block in the n^{th} frame
f_n^i	The original value of pixel i in frame n at the encoder
$\hat{f}_n^i$	The reconstructed value of pixel i in frame n at the encoder
$\hat{f}_{n\_UpsBase}$	The upsampled reconstructed frame n of the base spatial layer
$\hat{f'}_n^i$	The modified reconstructed value of pixel i in frame n at the encoder
$\hat{f}_n$	The reconstructed frame n at the encoder
$\hat{f'}_n$	The modified reconstructed frame n at the encoder
$\tilde{f}_n^i$	The reconstructed value of pixel i in frame n at the decoder
$\tilde{f'}_n$	The modified reconstructed value of pixel i in frame n at the decoder
J_{mode}	Cost function in the mode decision
$\tilde{K}_n^i$	The leaky value for the i^{th} pixel in the n^{th} frame at the decoder
$\hat{K}_n^i$	The leaky value for the i^{th} pixel in the n^{th} frame at the encoder
$L_{(n,m)}^{n+k}$	The set of blocks in frame $n+k$ which are using block m in frame n as a reference
n_ref0	Reference frame 0 for frame n
n_ref1	Reference frame 1 for frame n
p	Packet loss rate
PSR_i	The probability of successfully receiving the i^{th} packet
$\hat{r}_n^i$	The quantized prediction error of pixel i in frame n
R_C	Total channel bits
R_{mode}	The number of bits used in mode decision process
R_{motion}	The number of bits used in motion estimation process
R_S	Total source bits
R_{total}	The total bit budget
TEU	The total expected utility
Tk	The k^{th} temporal layers.
$TSD(n)$	The total sequence distortion when frame n is lost
TSD_0	The total sequence distortion when there was no loss
TSD_i	The total sequence distortion when the i^{th} NAL unit is lost
$TSN(n,m)$	The total sequence distortion when block m in frame n is lost
$U(n,m)$	The utility of block m in frame n
$U(n)$	The utility of frame n

$U^s(n)$	The utility of frame n, spatial layer s
U_i	The utility of packet i
w_0	The weight for reference 0 in the first proposed prediction structure

Chapter 1

Introduction

Due to the increasing demand for multimedia services over the last two decades, video coding has been an active research and standardization area [5–11]. In traditional video coding, compression efficiency was the most important requirement. The video files were either stored in recording devices or sent over networks. As the disk capacity was small and network bandwidth was restricted, improving the compression performance has been the most important issue. In other words, the major effort was focused on decreasing the video bandwidth in order to reduce the storage space on hard disks or the required network bandwidth to transmit it over networks without changing the quality too significantly.

With the rapid growth of technology, hard disks and networks are offering larger capacities and multimedia applications have attracted considerable attention. Applications like video conferencing, Internet video streaming, video on demand, mobile TV and high-definition TV broadcasting are widely used. New multimedia applications introduced new requirements in video coding. Today, in addition to compression performance, flexibility is an essential requirement. Modern video transmission systems use Internet and mobile networks which are usually based on RTP/IP [12]. Most RTP/IP access networks are heterogeneous environments where clients have different capabilities in terms of complexity, bandwidth, power and display resolution. The client devices might vary from cell phones with small screens and limited processing power to high definition TVs. In these environments, multiple clients with different requirements simultaneously access the same coded content. The scalable video coding concept was introduced to address the flexibility demands of multimedia applications in these environments.

In order to respond to requests of different clients, a video source needs to be coded multiple times. When the number of requests is limited, multiple time video coding is applicable. But, since video coding is a time consuming process, the increase in the number of clients makes this solution impractical. Scalable video coding (SVC) solves this problem by producing a flexible bitstream which accommodates different clients with different demands. This single bitstream contains the information to fulfil the requirement of different users. Clients can extract their required information from this stream easily. In other words, the main difference between SVC and single layer coding is that in SVC, multiple spatial (a wide range of resolutions), temporal (a wide range of frame rates) and quality (a wide range of quality levels) layers are provided while in single layer coding, the coded video has fixed resolution, frame rate and quality level.

Scalable video coding has been a research topic for more than 20 years, however previous scalable video coding standards have not attracted industrial attention and use, due to the significant performance loss and complexity of the decoding process introduced by scalable video coding [13]. The scalable extension of the H.264/AVC standard, which was finalized in 2007 [14], solves these two problems and outperforms all previous scalable video coding standards such as MPEG2 and MPEG4 [2]. One of the key features of this extension, known as scalable video coding (SVC), is compatibility with H.264/AVC [11], which is the latest video coding standard. The base layer is compatible with H.264/AVC and can be decoded by any H.264/AVC decoder. Most of the H.264/AVC components such as motion compensation, Intra prediction, transform, entropy coding, de-blocking and Network Abstraction Layer (NAL) unit packetization are used in SVC.

1.1 The Need for Error Resilience

A typical video communication system is shown in Fig. 1.1 [1]. The input video sequence is compressed at a desired bit rate by the source coder. The source coder can be divided into two separate parts: the waveform coder and the entropy coder. The waveform coder, which is a lossy device, compresses the video by using predictive coding, transform, like discrete cosine transform (DCT) or wavelet, and quantization. On the other hand, entropy coding converts the output of the waveform coder into a new representation using fewer bits in a lossless process. Entropy coding techniques like Huffman coding and arithmetic coding use the statistical characteristics of their inputs in order to compress them more efficiently.

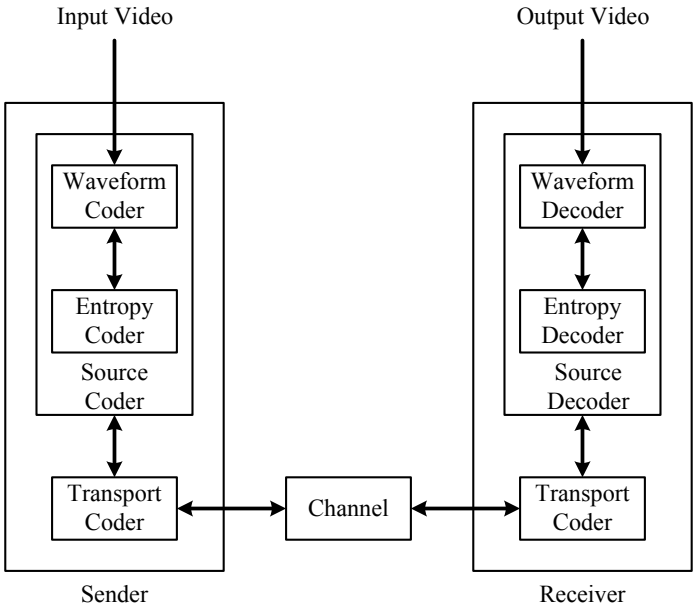


Fig. 1.1 A block diagram for a video communication system [1].

Examples of source coders include H.261 [6], MPEG-1 [7], MPEG-2 [8], H.263 [9], MEPG-4 [10] and H.264/AVC [11] codecs. The compressed video is encapsulated into proper transmission packets by the transport coder. The transport coding includes packetization, delivery policy selection, etc., and might vary based on the application. The transport packets are then transmitted over the channel. The reverse processes are performed at the receiver side and the transmitted video is reconstructed.

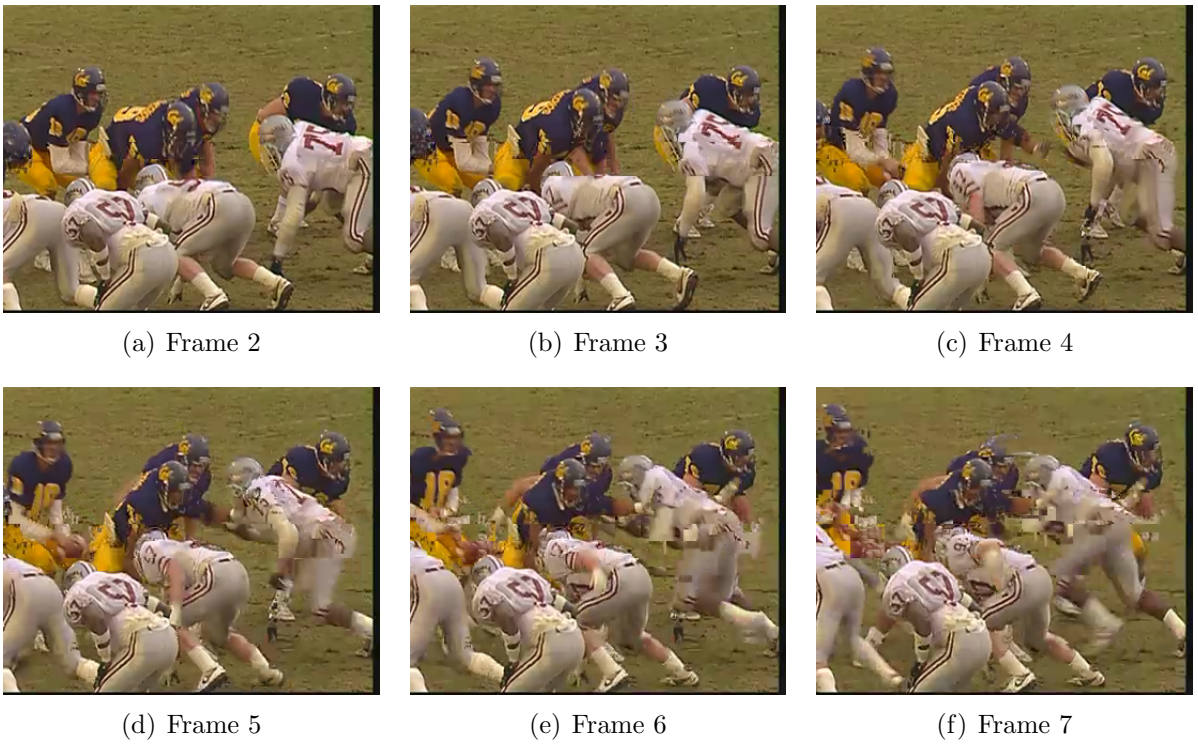
Video transmission is usually done over unreliable channels. In these channels like Internet and wireless networks, video packets may be lost. Also, due to the high bandwidth required for video transmission, quality of service (QoS) for video communication is not usually guaranteed in these networks. Over these channels, video packets may get discarded due to network congestion and buffer overflow at intermediate networks elements such as routers. Furthermore, due to playback requirements in some multimedia applications, delayed or out of order packets are also dropped.

On the other hand, in video coding process, predictive coding is used to achieve high compression performance by removing spatial and temporal redundancies. Intra prediction refers to the prediction of a pixel by using other areas of the same picture. Intra prediction removes the spatial redundancies. In order to remove the temporal redundancy in video

encoding, each frame uses previously encoded frames as reference for prediction. At the receiver side, the decoder is supposed to have the same reference frames in order to get the same video content. This process is referred to as Inter coding. When the transmission channel is error free, the encoder and the decoder references are synchronized. However, transmission of coded video over error prone channels is inevitable. Transmission errors result in quality degradation and due to the introduced mismatch between the encoder and the decoder, the error will possibly propagate to succeeding frames. As a result, the quality of the received video at the decoder side may drop significantly.

In order to provide a better representation of the error propagation effect, Fig. 1.2 depicts the error propagation effect in six consecutive frames of the standard "Football" test sequence transmitted over a channel with 10% packet loss rate (PLR). The video was encoded at 1000 kilobit per second (kbps) with common intermediate format (CIF) size and 15 frames per second (fps). In this channel, an error happened in frame 2. The lost area is concealed by copying from the co-located area in the previous frame. The sequence of frames shows that the damaged regions become larger in time. This is mainly due to using temporal prediction in coding of the input video. Temporal prediction utilizes previous frames as a reference, and if the reference is in error, it will be propagated to succeeding frames. The quality degradation is annoying, when this video is played at the decoder.

Due to the transmission of encoded video over error prone channels, building a video communication system that is robust to transmission errors has become an essential issue. In order to make the video transmission more robust, usually redundancy is added at the waveform coder, entropy coder or transport coder. The added redundancy is referred to as concealment redundancy [1]. Video coding techniques try to achieve high compression efficiency by removing redundancy while at the same time, adding concealment redundancy is required for handling packet losses and errors. Thus, there is a trade off between coding efficiency and error resilience under the constraint of the available bandwidth. For an error free case, all the available bandwidth is allocated to the source coding. As the channel error rate increases, more bit rate should be allocated to concealment redundancy to achieve the best quality at the receiver side. As a result, the goal of error resilient coding is to achieve the optimum point between video coding efficiency and error robustness.



(a) Frame 2 (b) Frame 3 (c) Frame 4

(d) Frame 5 (e) Frame 6 (f) Frame 7

Fig. 1.2 Subjective illustration of error propagation.

1.2 Classifying Error Resilience Techniques

Several techniques have been proposed to stop or decrease error propagation. As suggested in [1], these techniques are classified into three categories. This classification is based on the roles of the encoder and decoder in handling transmission errors. Forward error correction refers to techniques that add redundancy at the encoder side to make the bitstream more resilient. Error concealment by post-processing includes methods recovering the erroneous areas at the decoder by using the available information of neighbouring regions or previous frames. Finally, interactive error concealment covers techniques where the encoder and the decoder cooperate in order to minimize the effects of error propagation.

1.2.1 Forward error correction

The concealment redundancy can be added at the source or transport coder. One approach is to add redundancy at the source encoder by inserting more Intra macroblocks (MB), where a macroblock is a block of 16 x 16 pixels. Intra MBs prevent the propagation of

errors from previous frames. The inserted Intra MBs may be selected randomly [15, 16] or at specific places based on optimum techniques [17–20]. Moreover, conventional motion estimation and rate distortion optimization methods are designed to achieve maximum performance in error free channels. Several techniques have been proposed to modify these two parts of the encoder in order to satisfy both coding efficiency and error resilience [21–24]. Furthermore, various approaches estimate the end-to-end distortion in error prone channels. This distortion is used instead of source distortion in motion estimation or mode decision optimization to consider the state of channel [25–29]. Another approach is to change the prediction structure by modifying the reconstructed frame into a new one which is less vulnerable to transmission error. The modified reconstructed frame is used as reference in prediction of succeeding frames. These techniques are called reference frame modification techniques [30–34]. Some of these techniques are included in the video coding standards which will be presented in Chapter 2.

Another effective scheme for providing error resilience is protecting the coded sequence with forward error correction (FEC) codes. Since each video packet has different contribution to the video quality and different sensitivity to packet losses, unequal error protection (UEP) can be applied to video signals. The idea of UEP is to protect each video part differently based on its importance. Applying UEP on single layer video coding has been addressed by many researchers [35–37]. Using layered or scalable video coding in combination with unequal error protection achieves efficient robust video transmission [38–42]. Multiple description coding (MDC) [43] is another error resilience technique that produces two or more independent bitstreams, called descriptions. Each description is usually transmitted separately and can produce a basic quality. Since different descriptions are correlated, the coding efficiency of MDC is worse than single description coding. But the correlation can be used by the decoder in order to conceal the packet losses [44]. Generating proper descriptions in order to achieve efficient error concealment has been addresses by many researchers [45–48].

Error isolation by structure packetization is another approach which can be done at the transport level [49–51]. In this technique, the coded stream is packetized such that if a packet is lost, the rest of the packets can still be used. This is achieved by putting the header and coding modes in successive frames. In this way, the damaged areas are distributed in the frame and the error concealment can recover them easier. Error robust entropy coding techniques [52,53] which deal with bit errors also fall into this category. But

since our focus in this thesis is on packet oriented networks where handling of bit errors is not done by the source decoder, we will not review these techniques further.

1.2.2 Error concealment by post-processing

Error concealment by post-processing is a powerful tool used at the decoder side to recover the damaged area due to the transmission errors or losses. This tool, which does not require any additional redundancy, improves the quality of the received video. After the bitstream received at the decoder, it is examined for any error in the video syntax [54, 55]. If any error is detected, the error concealment tool would be used to conceal the loss. In a block based hybrid video coding, the error concealment might need to estimate the coding mode of the block, Inter or Intra, the texture information, including the DCT coefficient, the residual or the pixel values and the motion vectors for Inter coded blocks.

All error concealment techniques estimate the missing information by using the correlation between the corrupted blocks and their neighbouring blocks in the same frame or the previously received frames. If the information from the adjacent blocks within the frame is used, the technique is referred as spatial error concealment. Due to smoothness property of the video signals, some DCT coefficients in a corrupted block are likely to be close to the DCT coefficients of the neighbouring blocks. Different techniques have been proposed to exploit the spatial smoothness property. In [56], frequency domain interpolation is used to estimate the lost coefficients. In maximally smooth recovery [57], a number of DCT coefficient are estimated in order to achieve a smooth connection with the boundary pixels of the neighbouring blocks. Weighted pixel averaging can also be done in spatial domain [58–60]. These techniques work well for still images and Intra coded frames.

The basic idea in temporal error concealment techniques is to recover the damaged area by using previously received frames. In a simple method, a zero motion vector can be used for the lost block, which would result in concealing the block by copying the co-located area from the previous frame. This method is known as picture copy (PC). Recovery of coding mode and motion information would help the effectiveness of the error concealment [61,62]. In motion copy (MC) [63], the motion vectors and the coding modes are copied from the co-located block and motion compensation is done based on those. In another approach [64], the motion vector of the lost block is estimated by weighted averaging of the motion vectors of the neighbouring blocks. In more sophisticated methods [65, 66], spatial correlation and

frequency characteristics of still images are also used.

It should be noted that although error concealment techniques are powerful tools to improve the quality of the received video, these techniques increase the complexity of the decoding process which can limit their usability. Using very sophisticated techniques with high complexity is not possible for clients with computation and power constraints like mobile devices.

1.2.3 Interactive error concealment

In the last two error resilient categories, we discussed various methods from either the encoder side or the decoder side. For some applications, a backward channel from the decoder to the encoder is available. In these applications, interaction between the encoder and the decoder can potentially lead to the best performance. This is because the concealment redundancy can be added only when it is required. But due to application requirement and limitation, it might not be possible. Retransmission is a powerful interactive error concealment tool that may be used when the receiver can tolerate the delay of one transmission interval. Automatic retransmission on request (ARQ) techniques based on this concept have been developed [67–69]. However, due to the delay constraints in many multimedia application, the decoder cannot wait until it receives the requested packets. In order to solve the problem different solutions have been proposed.

In a simple method [70], the decoder would request for an Intra coded frame to stop the error propagation and continues decoding the next frames. The encoder might decide to encode an Intra frame or due to bandwidth limitation, Intra update the frame gradually. If gradual updating is selected, the encoder would make sure to only use Intra update area for prediction. This technique can be used in conversational video applications. Also, the decoder might inform the encoder about the received and lost frames. The encoder would use only the received frames as reference for future prediction in order to stop the error propagation of the erroneous frames [71]. In a more sophisticated technique [72, 73], the encoder keeps track of the propagation of the occurred errors. The damaged areas because of error propagation are not used for prediction of succeeding frames.

In this work, our main focus is on source level error resilience coding. A detailed review of some of these techniques are presented in Chapter 2.

1.3 Thesis Contributions

This thesis presents novel contributions in order to decrease the quality degradation caused by video transmission over error prone networks. All the proposed error resilience techniques are examined with the scalable extension of H.264/AVC. However, they are applicable to any other scalable video coding standard. The main contributions of this thesis are:

- We propose two new reference frame modification techniques that can be used in temporal and spatial scalability of SVC. Generally, modifying the conventional prediction structure is an approach to reduce the propagation of error to succeeding frames. In conventional prediction structure, the current reconstructed frame is used as a reference for the motion estimation and the motion compensation of the following frames. In this new approach, the reconstructed frame is modified into a new one which is less vulnerable to transmission error. The modified reconstructed frame is used as a reference in prediction of succeeding frames. Our first proposed technique improves the previous methods by exploiting the Intra MBs in reference frames efficiently [74]. The second proposed technique exploited a new leaky prediction structure in addition to efficiently making use of the Intra MBs in reference frames [75]. It jointly makes use of (i) error robustness of previous Intra MBs, (ii) good prediction resulting from using the previous reference frame, and (iii) exponential decay of error propagation caused by leaky prediction. It was observed that the video quality was increased especially for medium and high motion sequences.

- In conventional rate distortion optimization technique, only the source coding distortion is used for choosing the best block mode. As a result, only the best performance for error free channels is achieved. In order to satisfy both coding efficiency and error resilience in error prone channels, error resilience mode decision techniques use the end-to-end distortion instead of the source coding distortion in the rate distortion optimization. Various approaches have been proposed to estimate the end-to-end distortion in error prone channels. In order to get better performance, we combined the error resilience mode decision techniques with reference frame modification methods (RFM). However, using RFM techniques will change the prediction structures and new ways to estimate the end-to-end distortion are needed. In this thesis, the

end-to-end distortion of the second proposed prediction structure is calculated based on the LARDO technique [28]. By using the estimated distortion in the mode decision process, the best mode is selected based on compression efficiency and error robustness [76, 77].

- Another approach to address the problem of video transmission over error prone networks is unequal error protection (UEP) of scalable coded video. In this technique, different independent layers of an SVC stream are protected differently and based on their importance by using forward error correction (FEC) codes. Accurately analysing the importance or utility of each video part is a critical component and would lead to a better protection and higher quality of the received video. Calculation of the utility is usually based on multiple decoding of sub bitstreams and is highly computationally complex. In this work, we propose an accurate low complexity utility estimation technique that can be used in different applications. This technique estimates the utility of each network abstraction layer (NAL) by considering the error propagation to future frames. We utilize this method in an UEP framework with the scalable extension of H.264/AVC codec and we showed that it achieves almost the same performance as highly complex estimation techniques (an average loss of 0.05 dB). Furthermore, we propose a low delay version of this technique that can be used in delay constrained applications. The estimation accuracy and performance of our proposed technique are studied extensively [1].

1.4 Thesis Organization

In order to familiarize the reader with the topics covered in this dissertation, an extensive literature review of the related subjects is presented in Chapter 2. We begin by an overview of the scalable extension of H.264/AVC, known as SVC. It includes the history of scalable video coding in video coding standards, types of scalability in SVC, and other new introduced features of SVC. We proceed by studying the error resilience tools of H.264/AVC and its scalable extension. Then, rate distortion optimized error resilience techniques are discussed and the reader is introduced to the end-to-end distortion estimation methods. Finally, the reference frame modification techniques are reviewed.

[1] This work has been submitted to "Elsevier Signal Processing: Image Communication Journal"

In Chapter 3, we introduce two novel reference frame modification techniques that can be used in temporal and spatial scalability of SVC. Also, the end-to-end distortion of the new reference frame modification technique is presented. In this chapter, we begin by extending the existing reference frame modification schemes to the temporal and spatial scalability. We then proceed by introducing and explaining our proposed prediction structures and the reason they perform better than the existing schemes. We then focus on the end-to-end distortion calculation of our technique. Finally, we provide simulation results for all our proposed schemes showing improvements in performance.

Chapter 4 presents our low complexity utility calculation technique. It begins by presenting the existing utility estimation techniques. We then proceed by introducing and explaining our framework and the problem formulation. Then, the proposed utility calculation technique is introduced. Finally, we present simulation results for all our proposed techniques.

Chapter 5 summarizes the contributions of this dissertation and then, presents some possible directions for future work.

Chapter 2

Literature Review

In traditional video coding, compression efficiency was the most important requirement. With the rapid emergence of new technologies and demands, new multimedia applications with new requirements such as flexibility have been widely used. The scalable video coding concept was proposed to address this issue by generating a single bitstream that meets the requirement. An introduction to video coding is presented in Section 2.1. In Section 2.2, we will study the scalable video coding and specifically the scalable extension of H.264/AVC, known as SVC.

Furthermore, as the demand for new video services is growing rapidly, there is a considerable amount of coded video transmitted over error prone networks such as wireless networks or the Internet. In order to provide error robust video transmission, different techniques have been proposed. Each video coding standard provides some tools that improve the error resilience. Error resilience tools of H.264/AVC and its scalable extension are presented in Section 2.3. Most of these methods require adding redundancy to the coded stream that will compromise the coding efficiency. In order to make a reasonable trade off between coding efficiency and error resilience, rate distortion optimized error resilience techniques have been proposed which are discussed in Section 2.4. In Section 2.5, we study the end-to-end distortion estimation methods that can be exploited in finding the best coding modes and motion vectors with respect to the rate and the distortion. Another approach that can be employed independently of the above techniques is reference frame modification. In reference frame modification, the reference frame is modified to a new one which is less vulnerable to transmission error. These techniques are also reviewed in

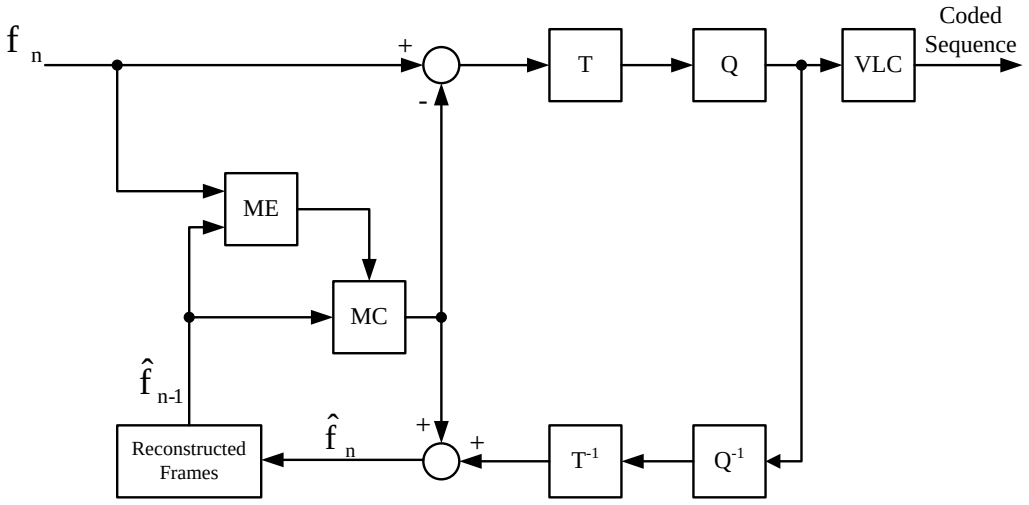


Fig. 2.1 Block diagram of a typical video encoder.

Section 2.6.

2.1 Video Coding

Various video coding standards, such as MPEG-2 [8] , MPEG-4 [10], H.263 [9], and H.264/AVC [11] employ a hybrid video coding framework. In hybrid video coding framework, the combination of a block based predictive coding and a transform coding is used. This combination would lead to effective video compression. The basic functional blocks are common among most of the video coding standards. Each standard would specify the syntax of the bitstream and the decoding process. This allows a flexibility in the design and implementation of the encoder, but the encoder should produce a standard decoder compliant bit stream.

The block diagram of a typical video encoder is shown in Fig. 2.1 where T, Q, Q^{-1} and T^{-1} denote transform, quantization, inverse quantization and inverse transform respectively. Also, variable length coding, motion estimation and motion compensation are represented by VLC, ME and MC respectively. This model contains two data paths: one going from left to right is the forward path, and the other going from right to left is the reverse path. Compression takes place within the forward path. Over the reverse path, decoding of the compressed video frame occurs. This entails that the encoder has a decoder

within itself.

In this diagram, the input frame is presented by f_n. Each frame is divided into blocks of 16x16 pixels. These blocks are called macroblocks (MBs). Each MB is coded separately as Intra or Inter block. Intra macroblocks are coded by removing the spatial redundancy of pixels within the current frame. A prediction signal is formed based on the information of the current frame. The difference between the input block and the prediction, which is referred to as residual, is then passed onto the transform block. The purpose of transform is to compact the energy of the residual signal. Most of the energy present in the residual video frame can be represented using only a few coefficients. The transformed coefficients are then quantized. The quantized coefficient are then reordered in a way that all the samples with significant values are grouped together. The reordered data is then entropy coded . The goal of this operation is to remove any statistical redundancy present in the data. To eliminate statistical redundancy, commonly occurring symbols are replaced with a shorter code and symbols occurring rarely are replaced with a longer code. To this end, variable length coder (VLC) or an arithmetic coder (AC) is used. The entropy-encoded code words together with side information required to decode the MB form the compressed stream. the side information needed to decode a macroblock are prediction mode, quantizer step size, motion vector information describing the location of the macroblock after motion compensation. Prediction, transform and entropy encoding are lossless process while quantization is lossy encoding.

Furthermore, pixels in successive frames are statistically related. This is referred to as temporal redundancy. In Inter macroblocks, the encoder exploits the temporal redundancy by using motion estimation and motion compensation. The previous coded frames are reconstructed and used as a reference to form the prediction by using motion estimation. These frames are called "reference frames" and are represented by $\hat{f}_n$ in Fig. 2.1. If one reference frame is used in motion estimation and motion compensation of the frame, it is referred to as a P frame, while if two frames are used, it is referred to as B frame. Motion estimation forms the prediction by finding a good match for the current MB in the reference frame. The residual is formed as the difference between the prediction and the input macroblock and is then transformed, quantized and coded by using variable length coding.

In the reverse path, the quantized coefficients go through the inverse quantization process and then through the inverse transform block to form the quantized residual. The

residual is then added to the prediction form the reconstructed frame ($\hat{f}_n$).

2.2 Scalable Video Coding

2.2.1 Scalable video coding in video coding standards

Early video coding standards such as ITU-T H.261 [6] and ISO/IEC MPEG-1 [7] were designed for specific applications such as storage and conversational service, which did not have scalability requirements. As a result these standards did not support scalability features. For parallel transmission or storage a method called simulcast was used. In simulcast method, two or more streams are put together for parallel transmission or storage. The first video compression standard which supported scalability was ISO/IEC MPEG-2 [8]. The main reason for adding this feature was the forward compatibility with MPEG-1. The base layer is encoded and decoded using the previous standard and the improved quality enhancement layer is encoded and decoded by the new one. In MPEG-2, as the enhancement data is encoded differentially with reference to the base layer, it cannot be used without the base layer. In other words, the base layer must be available to use the enhancement layer. All kinds of scalability (temporal, spatial and SNR) are supported in MPEG-2, but the number of layers is limited to three [4, 13].

The next video coding standard, MPEG-4 [10], supports more flexible scalability features. It also provides fine granular SNR scalability and video object level scalability. The basic approach of fine grain scalability (FGS) is re-quantization of coefficients in the discrete cosine transform (DCT) domain. It uses different quantization parameters for each layer. These parameters are larger for the base layer coding and, they decrease for each enhancement layers.

However, the scalability features of these standards were rarely employed. The main reasons were the significant loss in performance, and the complexity of the decoding process introduced by scalable video coding. In October 2003, the ISO/IEC Moving Picture Experts Group (MPEG) announced a call for proposal for a scalable video coding standard. In March 2004, 14 proposals were submitted and evaluated. In January 2005 MPEG and the ITU-T Video Coding Experts Group (VCEG) decided to jointly finalize the SVC project as an amendment of H.264/MPEG4-AVC standard [11]. The selected proposal provides the bitstream syntax and the decoding process. The reference encoding process is available

in the Joint Scalable Video Model (JSVM 11) [78]. SVC was finally standardized as an extension of H.264/AVC in 2007.

2.2.2 Types of scalability in scalable extension of H.264/AVC

Temporal scalability

In temporal scalability, the bitstream includes a temporal base layer and one or more temporal enhancement layers. Assuming T is the temporal layer identifier, $T = 0$ shows the base layer and $T = 1, 2, \ldots$ represent the higher enhancement layers. By removing all parts with T greater than a natural number k from the bitstream, a valid bitstream is obtained. This bitstream can be decoded by a SVC decoder.

Generally, if the motion-compensated prediction of a frame with a temporal layer identifier T is restricted to reference frames with temporal layer identifiers equal to or less than T, temporal scalability is achieved. Different levels of temporal scalability were supported in previous video coding standards such as MPEG-2, H.263, and MPEG-4. The reference picture memory control mechanism of H.264/AVC makes a more flexible temporal scalability available. Coding of picture sequences with different temporal dependencies is permitted. The only restriction is the maximum practical size of the Decoding Picture Buffer (DPB). Consequently, the temporal scalability of SVC was obtained by some minor changes in the signalling of temporal layers in H.264/AVC.

The concept of hierarchical B or P pictures, as shown in Fig. 2.2-a, leads to dyadic temporal enhancement layers. The numbers below the pictures show the coding order and Tk denotes the k^{th} temporal layers. The arrows represent the temporal prediction. The enhancement layer pictures are usually coded as B pictures. The two reference picture lists, *list0* and *list1*, of a picture with temporal layer identifier T are restricted to pictures with temporal layer identifiers less than T. As a result, the coded picture can be decoded without help of pictures with temporal layer identifiers greater than T. The described hierarchical structure is a special case and shows superior performance. It provides four temporal layers and structural delay of seven pictures [79].

The concept of multiple references of H.264/AVC can lead to other prediction structures. Fig. 2.2-b and Fig. 2.2-c are two examples of non-dyadic hierarchical structures. Fig. 2.2-b leads to three temporal layers and structural delay of eight pictures. The structure shown in Fig. 2.2-c has a delay of zero pictures and provides four temporal layers. It does not apply

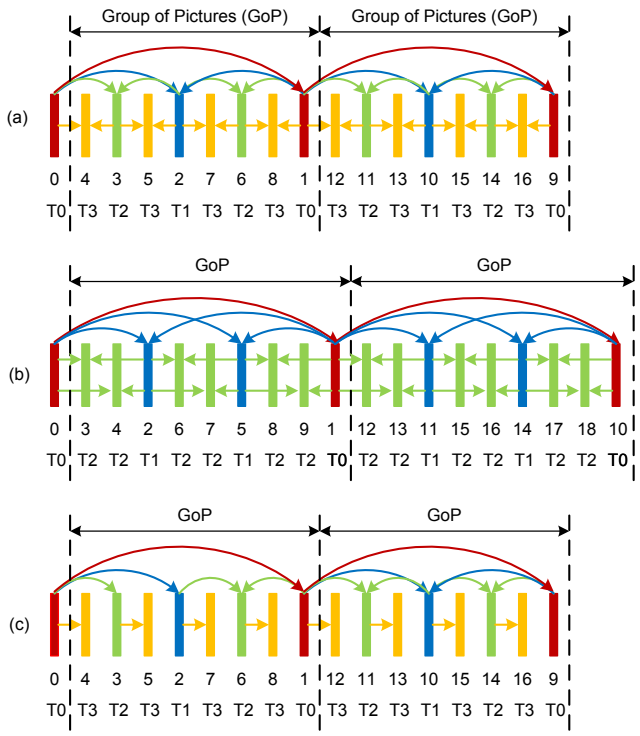


Fig. 2.2 Hierarchical prediction structures: (a) dyadic hierarchical prediction structure, (b) non-dyadic hierarchical prediction structure, (c) hierarchical prediction structure with zero delay [2].

motion-compensated prediction from upcoming pictures which leads to zero delay. The set of pictures between successive base layer pictures is referred as Group of Pictures (GoP). Selecting GoP size between 8 and 32 pictures usually achieves the best performance [2].

Spatial scalability

In previous video coding standards such as H.262, MPEG-2, H.263, and MPEG-4 spatial scalability was supported by multilayer coding. In SVC, the same approach is employed. Each spatial layer is recognized by using a dependency identifier D. $D = 0$ denotes the base spatial layer and $D = 1, 2, \ldots$ represent the higher spatial enhancement layers. In each spatial layer, the pictures are coded independently according to their layer motion parameters. However, the main difference with other video coding standards is the inter-layer prediction method. In this method, the encoder can use the base layer or other

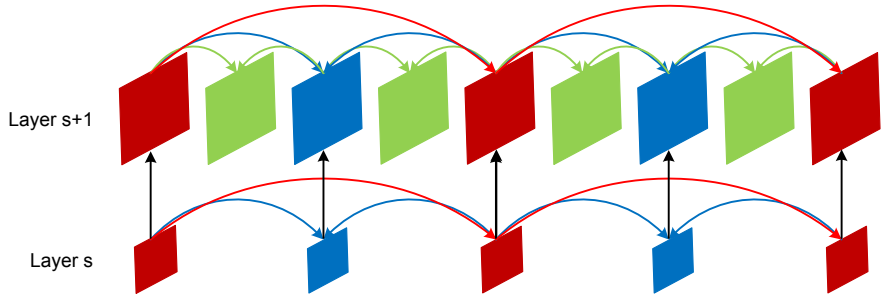


Fig. 2.3 Multi-layer structure with additional inter-layer prediction.

previous enhancement layers as reference for prediction. Inter-layer prediction structure can be seen in Fig. 2.3 in which vertical arrows represent inter-layer prediction.

By making use of lower layers information, the rate-distortion efficiency of the enhancement layers can improve significantly. The inter-layer prediction mechanisms in previous video coding standards were limited to using the reconstructed signals of the lower layers. In order to match the resolution of the enhancement layers the lower layers reconstructed signals are up-sampled. It is important to note that inter-layer prediction does not necessarily results in better performance comparing with the temporal prediction. The simulation results show that temporal prediction usually achieves better performance for slow or high spatial detail sequences [80]. Prediction of macroblock modes and motion parameters, and prediction of the residual signal are new inter-layer prediction modes added in SVC. They can improve the efficiency of spatial coding significantly [80].

- *Inter-Layer Motion Prediction:* A new type of macroblock is introduced in SVC in order to employ motion data of lower spatial layers. This macroblock type is used in spatial enhancement layers and is signaled by a syntax element called *base_mode_flag*. When this macroblock type is selected, additional information such as Intra-prediction modes and motion parameters is not transmitted. When the reference layer macroblock is inter-coded and *base_mode_flag* is equal to one, the enhancement layer macroblock is also inter-coded. Therefore, the partitioning of enhancement layer macroblock is obtained by scaling the co-located 8x8 block in the base layers. Motion vectors are also derived by upsampling the motion vectors in lower spatial layer.

- *Inter-Layer Residual Prediction:* In encoding Inter MBs, the residual signal of the

base layer can be used. A bilinear filter is used to up-sample the residual signal of the corresponding 8x8 blocks in the base layer. Then, the difference of the residual signal of the enhancement layer and up-sampled residual of lower layer is calculated, coded and transmitted. The difference signal usually has a smaller energy and requires fewer bits.

- *Inter-Layer Intra-Prediction:* The MBs that are encoded as Intra in the base layer can be up-sampled and used as reference for encoding MBs in higher layers. The corresponding block in the reference layer is up-sampled by using a one dimensional FIR filter for luma component and a bilinear filter for chroma components.

Quality scalability

Spatial scalability with the same resolution for the base and enhancement layer can be considered as quality scalability. This kind of quality scalability is called coarse-grain quality scalability (CGS). Almost all the concepts of spatial scalability can be employed in this case. As the base and enhancement layers have the same size, no upsampling is required in inter-layer prediction tools. The residual texture signal in the enhancement layer is requantized with a new quantization step size in order to achieve better quality. The quantization step size is smaller than the base and previous enhancement layer quantization step sizes. In order to decrease the decoding complexity, inter-layer Intra and residual prediction are directly done in the transform domain [80].

Use of CGS provides a scalable bitstream with a few limited bit rates. Each of these bit rates corresponds to a quality layer, so the number of supported bit rates is equal to the number of CGS layers. If the number of layers is increased and the quantization step sizes of different layers are close, the coding efficiency will decrease. The concept of medium-grain quality scalability (MGS) is introduced in SVC in order to provide a more flexible stream with a variety of bit rates. By using MGS, the number of achievable rate points is increased to 16 levels. Switching between MGS layers is allowed which will result in more flexible streams.

Since quality layers can be discarded at different points in the bitstream, reconstructed reference frames can differ between the encoder and the decoder. When references at the encoder and the decoder are not synchronized, there will be a quality degradation which is called drift. Different approaches for trading off enhancement layer coding efficiency and

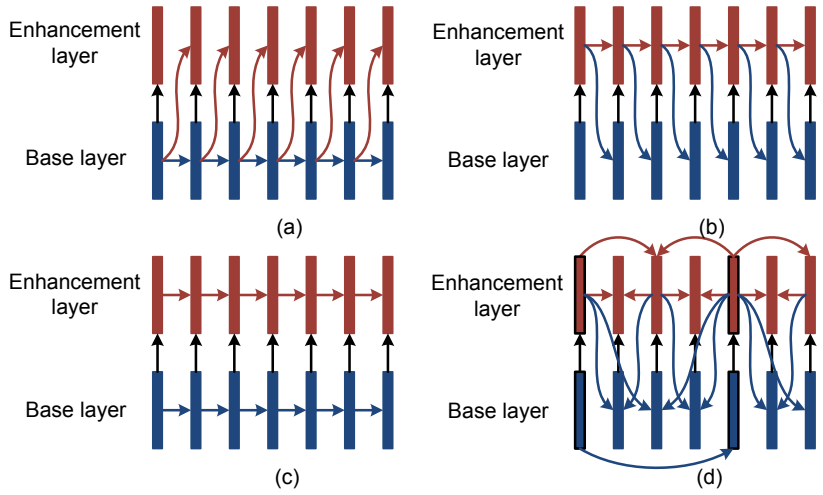


Fig. 2.4 Different methods for trading off enhancement layer coding efficiency and drift: (a) base layer control, (b) enhancement layer control, (c) two-loop control, (d) key picture concept of SVC [2].

drift are shown in Fig. 2.4 and described below.

- *Base layer only control:* In this scheme, motion compensation only employs the base layer reconstruction as reference. It is used for fine-grain quality scalability (FGS) in MPEG-4. Since any loss of a quality enhancement layer does not affect the motion compensation process, there is no drift in this method. On the other hand, the coding efficiency of enhancement layer will decrease significantly because of using a lower quality layer for prediction. This scheme is shown in Fig. 2.4-a.

- *Enhancement layer only control:* The other extreme case is using the highest available quality as reference for motion-compensated prediction. This method which is used in quality scalable coding of MPEG-2 is illustrated in Fig. 2.4-b. The advantage of this method is high coding efficiency of enhancement layer. Furthermore, since only one reference picture is stored for each time instant, complexity will decrease. The drawback of this scheme is the huge drift caused by any quality packet loss.

- *Two-loop control:* Another idea is using two independent motion compensation loops, one for the base layer prediction and the other for enhancement layer motion prediction. This concept is illustrated in Fig. 2.4-c. Spatial scalable coding in MPEG-2,

H.264, and MPEG-4 employ a similar concept. By using this method, quality refinement packet loss of enhancement layers will just lead to a drift in the enhancement layer.

- *SVC key picture concept:* A new approach using key pictures has been introduced in SVC. Key pictures are used as synchronization points, so drift propagation is limited between two key pictures. For the prediction of each key picture, only the base layer of the previous key picture is used as reference. Thus, there will not be any drift in these pictures. For the prediction of other pictures between two key pictures, the highest quality layer is used as reference. Consequently, the method has a high coding efficiency. This method is illustrated in Fig. 2.4-d. The black frame boxes represent key pictures. The concept of key picture can easily be combined with hierarchical prediction structures. All pictures in the base temporal layer are marked as key pictures.

2.2.3 Other features of SVC

Combined scalability

The basic concepts of temporal, spatial, and quality scalability in SVC can be combined. Fig. 2.5 illustrates the notion of combined scalability [3]. Each box represents a possible decodable video. An SVC bitstream may not have all kinds of scalability. Since quality and spatial scalability may cause a loss in coding efficiency, there is a trade-off between coding efficiency and degree of scalability. This trade-off is adjusted according to the application [81].

In SVC, a set of packets that correspond to one time are called an access unit. Temporal scalability is provided based on access units. Within each access unit, coding is structured in dependency layers. A dependency layer is a representation of a spatial resolution. CGS is an extreme case, in which the spatial resolutions of two dependency layers are the same. Each dependency layer may contain one or more quality layers. Each of the quality layers corresponds to the video at a specific time, with a specific resolution and a specific quality. Fig. 2.6 shows a typical encoder structure with three spatial layers. For each spatial layer, there is one independent encoder. The source video is up-sampled or down-sampled to

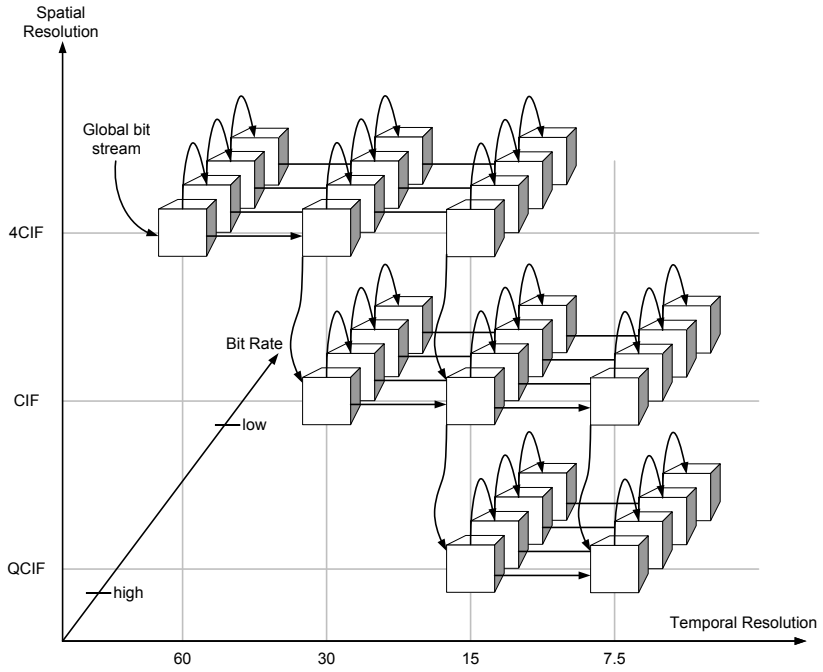


Fig. 2.5 Concept of flexible combined scalability [3].

match the required size of each spatial layer. The lowest layer is completely compatible with H.264/AVC encoder. For the higher layers, in addition to Intra layer coding, Inter-layer prediction is employed to remove the spatial redundancy and achieve better performance.

System interface

The one-byte header of Network Abstraction Layer (NAL) units in H.264/AVC is extended to three bytes in SVC. This will help to easily handle the bitstream of SVC. The dependency, quality, and temporal identifiers (D, Q, and T) together with other information are included in the extended header. Priority identifier P is one of the additional information which is included in the header. It indicates the importance of a NAL unit. An SVC bitstream may contain standard H.264/AVC NAL units which are called non-SVC NAL units. These NAL units do not have the SVC extended header unit, but as some of them are used in the SVC decoding process, prefix NAL unit are introduced. These NAL units which contain the SVC header extension go before all non-SVC NAL units. Another newly added NAL unit is Supplemental Enhancement Information (SEI) unit. SEI messages contain information

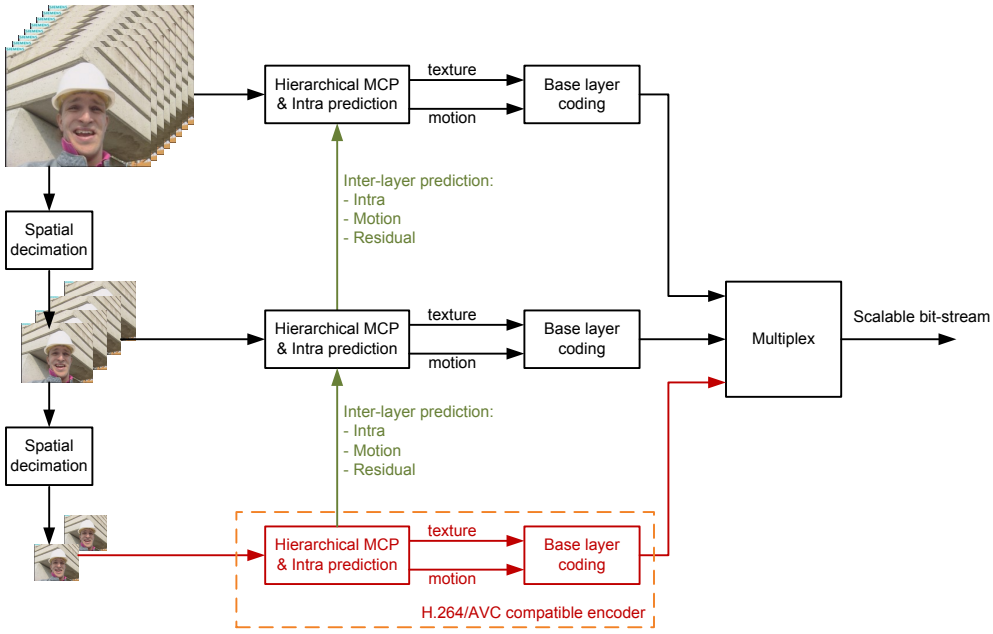


Fig. 2.6 SVC encoder structure example [2].

about spatial resolution and bit rates of the layers in the bitstreams [82].

Profiles

A set of coding tools that can be used in generating a bitstream is called a profile. Profiles help the inter-operability of applications with similar requirements. Three profiles are defined in SVC.

- *Scalable Baseline:* Low decoding complexity applications such as mobile broadcast, conversational and surveillance can make use of this profile. Base layer bitstream should follow the H.264/AVC baseline profile. But for coding the enhancement layers, B slices, weighted prediction, the CABAC entropy coding and the 8x8 luma transform can be employed. In addition, the resolution ratios of two following spatial layers are limited to 1.5 and 2.

- *Scalable High:* This profile is appropriate for broadcast, streaming, and storage applications. Base layer bitstream follow the high profile of H.264/AVC and there is no restriction on resolution ratios of two successive spatial layers.

- *Scalable High Intra:* This profile is designed for specific applications. In this profile, bitstreams contain only IDR pictures for all layers. Moreover, all the coding tools of the Scalable High profile are supported in this profile.

2.3 Error Resilience Tools for H.264/AVC and SVC

In video transmission systems, compressed video is delivered over channels that are not necessarily error free. Channel errors can lead to a mismatch in the encoder/decoder prediction loop which will propagate the errors to the succeeding frames. As a result, the quality of the received video at the decoder side may drop significantly. In order to reduce the introduced mismatch error resilience techniques have been proposed. Some of these techniques are included in the video coding standards. In this section, we highlight some of the more important error resilience tools in H.264/AVC and its scalable extension.

2.3.1 Intra updating

One approach to make the video stream resilient is to add redundancy at the encoder by inserting Intra macroblocks (MB). Intra MBs prevent the propagation of errors from previous frames. Encoding the entire frame as Intra is an extreme case that totally stops error propagation. This case is not practical in most of the applications due to bandwidth requirement. Inserting Intra MBs should be done in a proper way to result in an encoded stream with satisfactory coding efficiency and error resilience. Intra insertion methods can be classified to two categories. In the first group, there exists a mapping between the packet loss rate and Intra refreshing period. Intra MBs are inserted uniformly across the picture area. A simple method adds a definite number of Intra macroblocks randomly per picture [15]. Zhu et al. proposed a cyclic Intra refresh in which each MB is Intra updated within a given period of time [16].

The second group of algorithms performs Intra coding only for specific MBs in each frame. In [17], the Intra coded MBs are chosen based on the motion vectors. Regions with high and complicated motion are Intra refreshed. The proposed method in [18] tracks the motion of MBs in five future frames and counts the number of times each MB is used as a reference. A MB that is referenced more frequently is more likely to propagate errors. Thus, such a MB is encoded as Intra. The disadvantage of this method is an initial delay for computing first five frames. Based on the fact that people pay more attention to

some regions, a region of attention is defined [19]. This region has a higher priority to be coded as Intra. In [20], an Intra updating method is proposed that calculates the expected perceptual distortion of each MB by considering the human visual properties in addition to error sensitivity of the bitstream. The calculated distortion is used to select the Intra coded MBs. Both objective and subjective video quality improvements are reported.

Although inserting Intra MBs is one of the most basic and effective approaches to stop the error propagation, it does not consider the trade off between rate and distortion at the frame level. As a result, it does not achieve the optimum coding efficiency in combination with error robustness. Other techniques that consider both coding efficiency and error resilience are discussed in the following sections.

2.3.2 Multiple reference frames

H.264/AVC allows searching through multiple reference frames in order to find the best mode for each block. This feature improves the coding efficiency at the expense of increasing the computation and storage. This tool, which is added to the main profile, can help to improve the error resilience too. In a system with feedback feature, the encoder can be informed which frames have been received at the decoder through the feedback channel. These frames can be used as a reference for future prediction in order to stop the error propagation of the erroneous frames [83]. It has been reported that the performance of this scheme can even be better than inserting Intra refresh frames [84]. Furthermore, using multi-hypothesis motion compensated prediction (MHMCP), which was originally proposed to improve the coding efficiency [85], can be used as a error resilience tool [20,86–88]. In this technique, a linear combination of multiple references is used to predict a macroblock. Each of the references is called a hypothesis. For the case that one of hypotheses is damaged, all the error is not propagated due to averaging with other hypotheses. Also, the decoder might decide to cut prediction from the damaged hypothesis and use of the other one to make the prediction more reliable.

2.3.3 Picture segmentation

Each picture can be divided into one or more groups of macroblocks called slices. Each slice, which at least contains one macroblock, is coded independently of its neighbours. As a result, transmission errors will not propagate from one slice into another. Using slices will

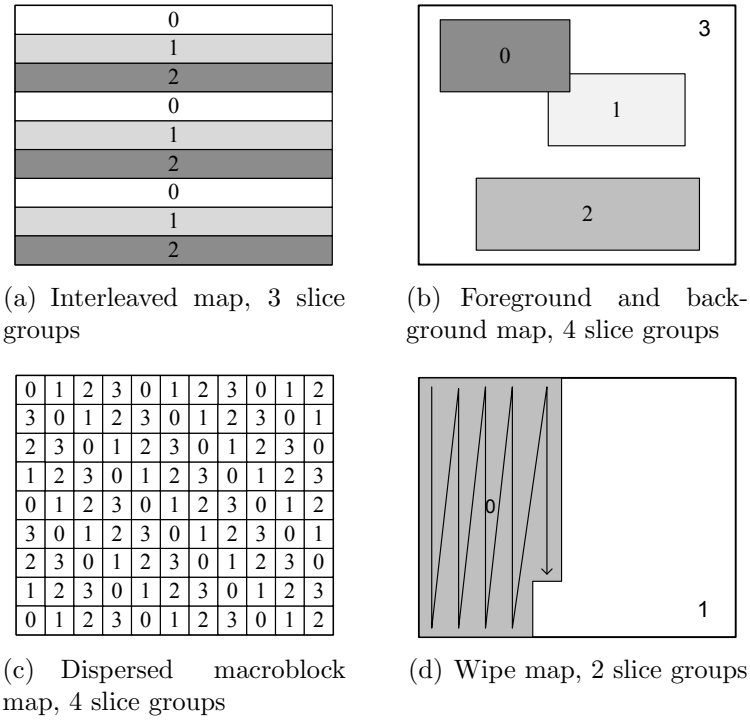
allow H.264/AVC to easily adapt to different network conditions. Increasing the number of slices would lead to small independent regions that helps error resilience. But on the other hand, it would decrease the coding efficiency because of the overhead of each slice or packet headers. For example, the packet header size of RTP/UDP/IP transmission is 40 bytes [89] which can significantly affect the efficiency of the coded video. In video transmission over wireless channels, each slice usually contains one row of macroblocks [19, 28]. In this way, if an error happens, only one row of the frame is corrupted rather than the entire picture. Furthermore, slice interleaving can be done in order to address the burst error problem [90]. Although slice interleaving would spread the burst errors, it would imply a delay which might not be tolerable in real time application.

2.3.4 Data partitioning

Data partitioning is another error resilience tool which is included in the H.264/AVC extended profile. In data partitioning, the slice data are placed in three different partitions A, B and C based on the importance:

- Partition A contains the most important data of the slice like the slice header and the header data for each macroblock including macroblock types, motions vector, etc.
- Partition B includes the residual data for MBs in I coded slices. Data in partition B is less important than partition A, but still more valuable than partition C.
- Partition C, which contains the least sensitive data, includes the residual data for MBs in P or B coded slices.

Each partition is placed in a separate NAL unit and might be transmitted independently. In addition, each partition has different importance and sensitivity to packet loss. Partition A is highly sensitive to transmission error. If Partition A is lost, Partition B and C are almost useless and can not be utilized. Also, since Intra blocks stop the error propagation from previous frames, Partition B is more important in error resilience aspect. As a result, applying unequal error protection on different partitions would improve the performance. Partition A is usually protected with the highest level, Partition B is protected with fewer extra bits and Partition C might not be protected at all [37, 91, 92]. Also, in some applications, each partition is transmitted over different channels with different reliability, using the most reliable one for partition A.



(a) Interleaved map, 3 slice groups

(b) Foreground and background map, 4 slice groups

(c) Dispersed macroblock map, 4 slice groups

(d) Wipe map, 2 slice groups

Fig. 2.7 FMO allocation maps [4].

2.3.5 Flexible macroblock ordering (FMO)

Macroblocks are usually placed in slices in raster scan order. By using flexible macroblock ordering (FMO), which is a feature in the baseline profile, macroblocks can be placed in slice groups in different allocation maps. Allocation maps are basically different spatial distribution of macroblocks within a frame and would not affect the coding process of a MB. Some of the allocation maps in the standard are shown in Fig. 2.7. Since each slice is decoded independently of the other slices, when a slice is lost, error would be distributed around the frame area. This would help the error concealment tool to recover the damaged area more effectively. As a result, using FMO can improve error resilience. In addition to macroblock allocation mappings specified in the standard, new mappings have been developed [93, 94] to improve the performance.

2.3.6 Redundant slices

Redundant slice allows the insertion of a redundant representation of a part or parts of a coded frame into the bitstream. At the decoder side, if there was no loss, the redundant slices are discarded and only the non-redundant slices are used for the decoding of the bitstream. In case of transmission errors, the decoder may replace the corrupted area in the frame by using the redundant slices. This tool, which is added to the baseline profile, improves the error resilience at the expense of spending more bits and slices. Since inserting redundant slices might decrease the coding performance, it is very important to properly select the redundant parts. For example, redundant slice might include a subset of macroblocks and different coding parameters. Different redundant slice selection have been proposed in [95, 96]. Also, multiple description schemes by using redundant slices has been proposed [45, 48].

2.3.7 Error resilience tools in SVC

All of the highlighted error resilience tools in H.264/AVC are also supported by SVC. In addition, three new standard error resilience tools are added in SVC:

- Quality layer integrity check signalling: This tool calculates a cyclic redundancy check (CRC) code from all the quality enhancement NAL units. This information is included in a supplemental enhancement information (SEI) NAL unit and sent to the decoder. The decoder can use this data to check if any of the quality enhancement NAL units are lost. If there is no loss, the encoder can use the highest quality layer as reference, which results in improved coding efficiency. If a loss is detected, the decoder may inform the encoder and the encoder would use the highest error free quality layer as reference. This will result in lower coding efficiency but will reduce the error propagation [97].

- Redundant picture property signalling: In order to indicate the correlations between a redundant representation and the corresponding non-redundant slice, this tool is used. In case of loss of the non-redundant slice, this information can be used by the decoder to use the redundant slice for:

  - Inter prediction or inter-layer prediction

  - Inter-layer mode prediction
  - Inter-layer motion prediction
  - Inter-layer residual prediction
  - Inter-layer texture prediction [98].

- Temporal level zero index signalling: Since temporal prediction is used in video coding, transmission error might propagate to future frames. The error propagation would damage more areas if it happens in the base temporal layer. The temporal level zero dependency representation index can be used to indicate which temporal base layer frame is used for encoding the current frame. By checking this index, the decoder would be able to determine if it has received all the frame in the lowest temporal layer. If the required frame is lost, the decoder can determine to send a feedback message or a retransmission request [99].

It should be mentioned that the discussed tools can be used individually or jointly to protect the compressed bitstream. These techniques do not change the encoding process of each block fundamentally. More details on these techniques and other standard tools can be found in [63, 100, 101].

2.4 Rate Distortion Optimized Error Resilient Techniques

Various approaches have been proposed to stop or decrease the impact of error propagation. As it was mentioned in Section 2.3.1, inserting Intra MBs is one of the most basic and effective approaches to stop the error propagation, but it does not consider the trade off between rate and distortion at the frame level. In this section, we mainly study the source coding error resilient techniques that achieve the optimum coding efficiency in combination with error robustness with respect to the available rate and distortion.

2.4.1 Error resilience mode decision methods

In video coding, each macrolock can be encoded as Inter, Intra or Skip mode. Furthermore, each MB can be divided into smaller blocks which results in having a choice among 17 different modes in the H.264/AVC standard [102]. The best mode should be selected in a

way that satisfies the main goal of an encoder, which is minimizing the total distortion D, where rate R is subject to a bit constraint R_{Target}, for each MB:

$$\min\ D\ \text{ subject to } R\ <\ R_{\text{Target}}. \tag{2.1}$$

This problem is typically solved using Lagrangian optimization [103] where the best mode is selected in such a way that the Lagrangian cost function is minimized. This cost function is defined as:

$$J_{\text{mode}}\ = D + \lambda_{\text{mode}} R_{\text{mode}}. \tag{2.2}$$

D and R_{mode} respectively denote the distortion between the original and the reconstructed MB and the number of bits for coding the prediction residue, selected motion vectors, and MB header corresponding to the mode. λ_{mode} depends on the quantization parameter (QP) and is computed as [103]:

$$\lambda_{\text{mode}} = 0.85 \times 2^{(QP-12)/3}, \tag{2.3}$$

In this rate-distortion optimization technique, the source distortion is only considered for selecting the best mode. Therefore, it achieves the best performance for error free channels. Several researchers modified this method in order to satisfy both coding efficiency and error resilience. The proposed methods are known as error resilience mode decision methods.

In [21], first, costs of the best Inter mode (J_{Inter}) and the best Intra mode (J_{Intra}) are calculated. Then, if the ratio of Intra cost over Inter cost ($J_{\text{Intra}}/J_{\text{Inter}}$) is smaller than a constant, Intra mode is selected, otherwise the MB is encoded as Inter. The constant is calculated experimentally and is set to 1.3. This work was further improved in [22] by proposing a mapping function to compute the constant. The constant is computed as a function of packet loss and bit rates.

A weighted distortion mode decision technique is proposed in [23]. In this method, the mode decision cost function is modified by adding a weighting factor (w_{mode}):

$$J_{\text{mode}}\ = w_{\text{mode}}\ D + \lambda_{\text{mode}} R_{\text{mode}}. \tag{2.4}$$

This factor is computed by running a two-phase encoding process. In the first phase,

motion prediction information of each MB in future frames is collected. This information includes the number of pixels of each MB used in succeeding frames. In the second encoding phase, the MBs that are more frequently used as reference for succeeding frames take a higher factor. As a result, these MBs are more likely to be encoded as Intra.

Using the estimated end-to-end distortion, instead of the source distortion in Eq. (2.2) has been addressed by many researchers [25, 26, 28, 29, 104]. These techniques find the best modes for each MB by taking into consideration both source and channel distortion. In order to improve these methods further, they can be used in cooperation with other techniques. For instance, in addition to MB mode, the motion vectors can be selected in a way that consider the channel errors.

2.4.2 Error resilience motion estimation methods

Motion compensated prediction (MCP) is a major component in all video coding standards. It removes the temporal redundancy between consecutive frames efficiently. In conventional video coding standards, the best motion vector is selected by rate-distortion optimized motion estimation through minimizing the Lagrangian cost function [105]. The cost function is defined as:

$$J_{\text{motion}} = D_{\text{motion}} + \lambda_{\text{motion}} R_{\text{motion}}, \tag{2.5}$$

R_{motion} denotes the number of bits required for coding the motion vector. The distortion measure can be calculated as Sum of Absolute Differences (SAD) or Sum of Square Differences (SSD) between the original and the matched block in the reference frame. Based on the selected distortion measure, λ_{motion} is computed; $\lambda_{\text{motion}} = \lambda_{\text{mode}}$ when SSD is used and $\lambda_{\text{motion}} = \sqrt{\lambda_{\text{mode}}}$ when SAD is the distortion measure. λ_{mode} is calculated as in Eq. (2.3).

Similar to the conventional mode decision optimization, the motion estimation optimization method is proposed for error free channels and is not suitable for channels with errors. A number of researches have been conducted that consider the channel errors in motion estimation. In [24], a method is proposed to reduce the error propagation by predicting from safe areas. Safe areas are defined as regions with lower chance of error spread such as recently Intra updated MBs. In order to bias the motion vector toward these regions a weight factor (w) is added to Eq. (2.5).

$$J_{\text{motion}} = wD_{\text{motion}} + \lambda_{\text{motion}} R_{\text{motion}}, \tag{2.6}$$

w takes a value based on the temporal distance from the last Intra updated MB. A more recently Intra refreshed block has a lower weight factor and is more probable to be selected as the reference block.

In [106], instead of using the source distortion, the end-to-end distortion is used in RD optimization. The end-to-end distortion is calculated by ROPE [26] (See Section 2.5.3) as Eq. (2.10). Although the proposed method achieves significant performance improvements, it requires a huge amount of computation and storage resources due to pixel level distortion estimation. A simpler technique is proposed in [107] that estimates the expected end-to-end distortion by only calculating the first moment of each pixel at the decoder. This technique uses SAD as the measure and categorizes the distortion as source and channel distortion. The selected motion vector minimizes the total expected distortion.

2.5 End-to-end Distortion Estimation

As it was stated in the previous section, by considering the end-to-end distortion instead of source distortion, both coding efficiency and error resilience are satisfied. Various approaches have been proposed to estimate the end-to-end distortion in error prone channels. The distortion can be estimated at the pixel level, the block level or at the frame level. In the following, we will study the most important end-to-end distortion estimation techniques.

Notation

Throughout this thesis, we will refer to the i^{th} pixel in the n^{th} frame of the original sequence as f_n^i. $\hat{f}_n^i$ and $\tilde{f}_n^i$ denote the reconstructed values of pixel i in frame n at the encoder and decoder, respectively. Also, we will define different types of distortion in following sections. $d_{\cdot}(n,i)$ refers to as the distortion at the pixel level for frame n and pixel i. "$\cdot$" can be replaced by different types of distortion. The distortion at block level for the m^{th} block in the n^{th} frame is defined as $D_{\cdot}(n,m)$ and $D_{\cdot}(n)$ denotes the distortion of frame n. By using this notation, we will have:

$$D_{\cdot}(n) = \sum_{m=1}^{M} D_{\cdot}(n,m),$$

$$D_{\cdot}(n,m) = \sum_{i \in B_m} d_{\cdot}(n,i),$$

where M is the number of blocks in one frame and B_m is defined as the set of pixels in block m.

2.5.1 Block weighted distortion estimate (BWDE)

This technique [108] is one of the earliest techniques that estimate the end-to-end distortion on a MB basis. The distortion at the MB level for frame n and MB m is computed as:

$$D(n,m) = (1-p)D_1(n,m) + pD_2(n,m), \tag{2.7}$$

where

$$D_1(n,m) = D_{\text{src}}(n,m) + \sum_{l=1}^{L} pD_2(n-l,m). \tag{2.8}$$

$D_{\text{src}}(n,m)$ is the source coding distortion and L represents the number of frames since the last Intra coded frame. $D_2(n,m)$ denotes the weighted average of the concealment distortion of the MBs in the previous frame that are mapped to the current block by motion compensation. The weighting is based on the covered area. $D_2(n,m)$ is stored per MB and used for the computation of $D_1(n,m)$ in succeeding frames. It should be noted that since for an Intra block there is no temporal prediction, $D_1(n,m)$ is equal to the source coding distortion. p denotes the packet loss rate.

Since this technique does not consider the error propagation related to temporal error concealment, the estimated distortion is not very accurate. Also, it is assumed that the employed error concealment technique at the decoder is know at the encode time.

2.5.2 K-Decoders

Error robust rate distortion optimization (ER-RDO) [25] estimates the end-to-end distortion of a MB as the average distortions of the MB over K different random variable channel realizations ($C(k)$). It relies on implementing K decoders at the encoder side. The

end-to-end distortion the i^{th} pixel in frame n is estimated as:

$$d(n,i) = \frac{1}{K}\sum_{k=1}^{K}\left|f_n^i - \tilde{f}_n^i(C(k))\right|^2 \tag{2.9}$$

where f_n^i and $\tilde{f}_n^i$ denote the original value at the encoder and the reconstructed value at the decoder of pixel i in frame n. Setting K to a large value (> 100) leads to an accurate estimation; on the other hand, increasing the value of K imposes high computation complexity and massive storage requirements which are not practical in all applications. It has been suggested that $K = 30$ is proper for most of applications and in order to achieve very accurate results, $K = 500$ can be used [25]. This method has been adopted in the H.264/AVC test model [109], and is referred to in this thesis as ER-RDO or K-Decoders.

2.5.3 ROPE

Recursive optimal per-pixel estimate (ROPE) [26] is another algorithm that computes the end-to-end distortion at the pixel level. Assuming the reconstructed pixel at the decoder ($\tilde{f}_n^i$) is a random variable, the end-to-end distortion of the i^{th} pixel in the n^{th} frame ($d(n,i)$) is defined as:

$$d(n,i) = E\left\{\left(f_n^i - \tilde{f}_n^i\right)^2\right\} \tag{2.10}$$

$$= \left(f_n^i\right)^2 - 2f_n^i E\left\{\tilde{f}_n^i\right\} + E\left\{\left(\tilde{f}_n^i\right)^2\right\}, \tag{2.11}$$

where f_n^i and $\tilde{f}_n^i$ denote the original value at the encoder and the reconstructed value at the decoder of pixel i in frame n. In order to find the distortion, the first and second moments of each pixel must be calculated. It is assumed that packet loss occurrences are independent [110] of the packet loss rate p. Based on whether the pixel is Intra or Inter coded, two different cases are considered:

Pixel in an Intra coded MB

For these pixels, three different cases are considered:

1. Each packet is received correctly with probability $1 - p$. In this case, $\tilde{f}_n^i$ is equal to

$\hat{f}_n^i$, where $\hat{f}_n^i$ represents the reconstructed pixel at the encoder.

2. When the current packet is lost, the decoder checks the previous packet. If the previous packet has arrived correctly, the median motion vector of the nearest MBs is calculated and used for the concealment of the lost pixel. In this case, which occurs with a probability of $(1-p)p$, $\tilde{f}_n^i$ is equal to $\tilde{f}_{n-1}^l$, where l is the location of the pixel used for concealment.

3. If the current and previous packets are both lost, it is assumed that the lost MB is concealed by copying the co-located MB from the previous frame ($\tilde{f}_{n-1}^i$). This event occurs with a probability of p^2.

   The first and second moment of pixel i of frame n are calculated as:

$$E\left\{\tilde{f}_n^i\right\} = (1-p)\hat{f}_n^i + (1-p)pE\left\{\tilde{f}_{n-1}^l\right\} + p^2E\left\{\tilde{f}_{n-1}^i\right\}. \tag{2.12}$$

$$E\left\{\left(\tilde{f}_n^i\right)^2\right\} = (1-p)\left(\hat{f}_n^i\right)^2 + (1-p)pE\left\{\left(\tilde{f}_{n-1}^l\right)^2\right\} + p^2E\left\{\left(\tilde{f}_{n-1}^i\right)^2\right\}. \tag{2.13}$$

Pixel in an Inter coded MB

Assuming that pixel j in frame $n-1$ is the reference for pixel i in frame n and $\hat{r}_n^i$ represents the quantized prediction error we have:

$$\hat{r}_n^i = \hat{f}_n^i - \hat{f}_{n-1}^j.$$

When a pixel is not lost, the motion vector (MV) and the residue are received correctly at the decoder. Thus, the decoder reconstructed pixel is $\tilde{f}_n^i = \hat{r}_n^i + \tilde{f}_{n-1}^j$. In the case of packet loss, the decoder performs the same operation as for Intra pixels. The first and second moment of pixel i of frame n are calculated as:

$$\begin{aligned} E\left\{\tilde{f}_n^i\right\} &= (1-p)\left(\hat{r}_n^i + E\left\{\tilde{f}_{n-1}^j\right\}\right) + (1-p)pE\left\{\tilde{f}_{n-1}^l\right\} + p^2E\left\{\tilde{f}_{n-1}^i\right\}. \qquad (2.14)\\ E\left\{\left(\tilde{f}_n^i\right)^2\right\} &= (1-p)\left(\hat{r}_n^i + E\left\{\tilde{f}_{n-1}^j\right\}\right)^2 + (1-p)pE\left\{\left(\tilde{f}_{n-1}^l\right)^2\right\} + p^2E\left\{\left(\tilde{f}_{n-1}^i\right)^2\right\} \\ &= (1-p)\left((\hat{r}_n^i)^2 + 2\,\hat{r}_n^i\,.E\left\{\tilde{f}_{n-1}^j\right\} + E\left\{\left(\tilde{f}_{n-1}^j\right)^2\right\}\right) \end{aligned}$$

$$+(1-p)pE\left\{\left(\tilde{f}_{n-1}^{l}\right)^{2}\right\}+p^{2}E\left\{\left(\tilde{f}_{n-1}^{i}\right)^{2}\right\}. \tag{2.15}$$

By using Eq.(2.12) - Eq.(2.15), the encoder can recursively update the first and the second moments based the available information. The ROPE method was initially proposed to perform at full pixel level, but later in [27, 111], an extended version was proposed that works at sub pixel level. Estimating the end-to-end distortion in DCT domain and error resilience rate control based on ROPE were proposed in [112] and [113]. Comparing to ER-RDO with ($K = 100$), ROPE requires less computational resources, but it is still complex in terms of computation and storage [28].

2.5.4 LARDO

Loss aware rate distortion optimization (LARDO) [29] is another recursive technique that estimates the end-to-end distortion. It classifies the distortion as source coding (d_{src}), error propagation (d_{ep}), and error concealment (d_{ec}) distortions. It works based on the assumption that if a block is received at the decoder with no loss, the distortion of the block is calculated as the sum of the source coding distortion and the error propagation distortion from the block used as reference to predict the current block. For Intra coded blocks, error propagation distortion is equal to zero. If the block is lost, the total end-to-end distortion is equal to the error concealment distortion. So, assuming a packet loss rate of p, and

$d_{\text{src}}(n,i)=E\left\{\left(f_{n}^{i}-\hat{f}_{n}^{i}\right)^{2}\right\}$, i.e., the distortion due to compression, which can be computed at the encoder,

$d_{\text{ep}}(n,i)=E\left\{\left(\hat{f}_{n}^{i}-\tilde{f}_{n}^{i}\right)^{2}\right\}$, i.e., the mean square difference between the encoder and decoder reconstruction of pixel i in frame n, and due to error propagation,

$d_{\text{ec}}(n,i)=E\left\{\left(f_{n}^{i}-\tilde{f}_{n-1}^{l}\right)^{2}\right\}$, i.e., the mean square difference between a pixel and the pixel used to conceal it when it is lost, (In order to be adaptable to different concealment methods, it is assumed that pixel i is concealed by copying pixel l of frame $n-1$. In case of performing simple picture copy for concealment, l is equal to i.)

the end-to-end distortion of pixel i in frame n is computed as:

$$\begin{aligned} d(n,i) &= E\Big\{\Big(f_n^i - \tilde{f}_n^i\Big)^2\Big\} \\ &= (1-p)E\Big\{\Big(f_n^i - (\tilde{f}_{ref}^j + \hat{r}_n^i)\Big)^2\Big\} + pE\Big\{\Big(f_n^i - \tilde{f}_{n-1}^l\Big)^2\Big\} \\ &= (1-p)E\Big\{\Big(f_n^i - (\tilde{f}_{ref}^j + \hat{f}_n^i - \hat{f}_{ref}^j)\Big)^2\Big\} + pE\Big\{\Big(f_n^i - \tilde{f}_{n-1}^l\Big)^2\Big\} \\ &= (1-p)E\Big\{\Big(f_n^i - \hat{f}_n^i\Big)^2\Big\} + (1-p)E\Big\{\Big(\hat{f}_{ref}^j - \tilde{f}_{ref}^j\Big)^2\Big\} + pE\Big\{\Big(f_n^i - \tilde{f}_{n-1}^l\Big)^2\Big\} \quad (2.16) \\ &= (1-p)\Big(d_{\text{src}}(n,i) + d_{\text{ep}}(ref,j)\Big) + p\; d_{\text{ec}}(n,i), \quad (2.17) \end{aligned}$$

where $\hat{f}_n^i$ and $\tilde{f}_n^i$ denote the reconstructed value of pixel i at the encoder and at the decoder respectively. The reference of i^{th} pixel in the n^{th} frame is pixel j in the ref^{th} frame, and the quantized prediction error is denoted by $\hat{r}_n^i$. Also, Eq. (2.16) is based on the assumption that effects of source distortion at the encoder and error propagation at the decoder are additive.

Source coding distortion is the distortion caused by quantization and is calculated as the mean square error (MSE) between the original pixel and the reconstructed pixel. Since the original and reconstructed pixels are available at the encoder side, the source coding distortion can be promptly calculated.

Error concealment distortion is the distortion caused by applying error concealment on the lost blocks. This distortion might be different based on the utilized error concealment technique. The more complicated the error concealment technique used, the lower the error concealment distortion. $d_{\text{ec}}(n,i)$ is calculated as:

$$\begin{aligned} d_{\text{ec}}(n,i) &= E\Big\{\Big(f_n^i - \tilde{f}_{n-1}^l\Big)^2\Big\} \\ &= E\Big\{\Big(f_n^i - \hat{f}_{n-1}^l + \hat{f}_{n-1}^l - \tilde{f}_{n-1}^l\Big)^2\Big\} \\ &= E\Big\{\Big(f_n^i - \hat{f}_{n-1}^l\Big)^2\Big\} + E\Big\{\Big(\hat{f}_{n-1}^l - \tilde{f}_{n-1}^l\Big)^2\Big\} \quad (2.18) \\ &= d_{\text{ec\_org}}(n,i) + d_{\text{ep}}(n-1,k), \quad (2.19) \end{aligned}$$

where $d_{\text{ec\_org}}(n,i) = E\Big\{\Big(f_n^i - \hat{f}_{n-1}^l\Big)^2\Big\}$ is the original frame error concealment distortion,

which is the MSE between the original and potential error concealment pixels and is available at the encoder side. $d_{\text{ep}}(n-1,k)$ is the error propagation from the previous frame. Also, Eq. (2.18) is based on the assumption that error concealment distortion at the encoder and error propagation in the decoder are additive. Based on the assumption that the error concealment technique is known at the encoder side, error concealment can be calculated inside the encoder. The error propagation distortion represents the distortion that propagates to future frames based on the prediction structure and is calculated as:

$$
\begin{aligned}
d_{\text{ep}}(n,i) &= E\Big\{\Big(\hat{f}_n^i - \tilde{f}_n^i\Big)^2\Big\} \\
&= (1-p)E\Big\{\Big(\hat{f}_n^i - (\tilde{f}_{ref}^j + \hat{r}_n^i)\Big)^2\Big\} + pE\Big\{\Big(\hat{f}_n^i - \tilde{f}_{n-1}^l\Big)^2\Big\} \\
&= (1-p)E\Big\{\Big(\hat{f}_n^i - (\tilde{f}_{ref}^j + \hat{f}_n^i - \hat{f}_{ref}^j)\Big)^2\Big\} + pE\Big\{\Big(\hat{f}_n^i - \tilde{f}_{n-1}^l\Big)^2\Big\} \\
&= (1-p)E\Big\{\Big(\hat{f}_{ref}^j - \tilde{f}_{ref}^j\Big)^2\Big\} + pE\Big\{\Big(\hat{f}_n^i - \tilde{f}_{n-1}^l\Big)^2\Big\} \\
&= (1-p)E\Big\{\Big(\hat{f}_{ref}^j - \tilde{f}_{ref}^j\Big)^2\Big\} + pE\Big\{\Big(\hat{f}_n^i - \hat{f}_{n-1}^l + \hat{f}_{n-1}^l - \tilde{f}_{n-1}^l\Big)^2\Big\} \\
&= (1-p)E\Big\{\Big(\hat{f}_{ref}^j - \tilde{f}_{ref}^j\Big)^2\Big\} + pE\Big\{\Big(\hat{f}_n^i - \hat{f}_{n-1}^l\Big)^2\Big\} + pE\Big\{\Big(\hat{f}_{n-1}^l - \tilde{f}_{n-1}^l\Big)^2\Big\} \quad (2.20) \\
&= (1-p)d_{\text{ep}}(ref,j) + p\Big(d_{\text{ec\_rec}}(n,i) + d_{\text{ep}}(n-1,k)\Big), \quad (2.21)
\end{aligned}
$$

where $d_{\text{ec\_rec}}(n,i) = E\Big\{\Big(\hat{f}_n^i - \hat{f}_{n-1}^l\Big)^2\Big\}$ is the reconstructed frame error concealment distortion, which is the MSE between the reconstructed and error concealment pixel and is also available at the encoder side. Eq. (2.20) is also based on the assumption that error concealment distortion at the encoder and error propagation in the decoder are additive. By assuming constraint Intra prediction, which means Intra prediction is only done by using the neighbouring Intra coded pixels, Intra blocks have zero error propagation.

In order to reduce the storage complexity, the error propagation distortion is estimated recursively per pixel but stored per block for each frame. The stored values are used to calculate the error propagation to future frames. At the block level, the end-to-end distortion of block m in frame n ($D(n,m)$) is computed as:

$$
D(n,m) = (1-p)\Big(D_{\text{src}}(n,m) + D_{\text{ep}}(ref, ref(m))\Big) + p\ D_{\text{ec}}(n,m). \quad (2.22)
$$

where source, error propagation and error concealment distortions of block m in the n^{th} frame are represented by $D_{\text{src}}(n,m)$, $D_{\text{ep}}(n,m)$ and $D_{\text{ec}}(n,m)$ respectively, and calculated as:

$$\begin{aligned} D_{\text{src}}(n,m) &= \sum_{i \in B_m} d_{\text{src}}(n,i), \\ D_{\text{ep}}(n,m) &= \sum_{i \in B_m} d_{\text{ep}}(n,i), \\ D_{\text{ec}}(n,m) &= \sum_{i \in B_m} d_{\text{ec}}(n,i). \end{aligned}$$

B_m is defined as the set of pixels in block m and $ref(m)$ represents the block used for prediction of block m in frame n. Assuming l is the block used for error concealment of block m, $D_{\text{ec}}(n,m)$ is calculated as:

$$D_{\text{ec}}(n,m) = D_{\text{ec\_org}}(n,m) + D_{\text{ep}}(n-1,l), \tag{2.23}$$

where $D_{\text{ec\_org}}(n,m) = \sum_{i \in B_m} d_{\text{ec\_org}}(n,i)$ and the error propagation distortion is calculated as:

$$D_{\text{ep}}(n,m) = (1-p)D_{\text{ep}}(ref, ref(m)) + p\Big(D_{\text{ec\_rec}}(n,m) + D_{\text{ep}}(n-1,K)\Big), \tag{2.24}$$

where $D_{\text{ec\_rec}}(n,m) = \sum_{i \in B_m} d_{\text{ec\_rec}}(n,i)$ is the reconstructed frame error concealment distortion of the m^{th} block in the n^{th} frame.

The distortion estimation of LARDO was reported as accurate as ROPE and ER-RDO with $K = 100$ for H.264/AVC but with much less computational complexity [28]. Furthermore, LARDO finds the optimum modes by using Lagrangian method by considering the accurately estimated end-to-end distortion. LARDO has been extended to multi-layer coding for SVC [114] and implemented in the SVC reference software, Joint Scalable Video Model (JSVM) [78]. We will be using and modifying this framework in subsequent chapters.

It should be mentioned that all these end-to-end distortion calculation techniques require to estimate the packet loss rate accurately, which might not be available in many applications.

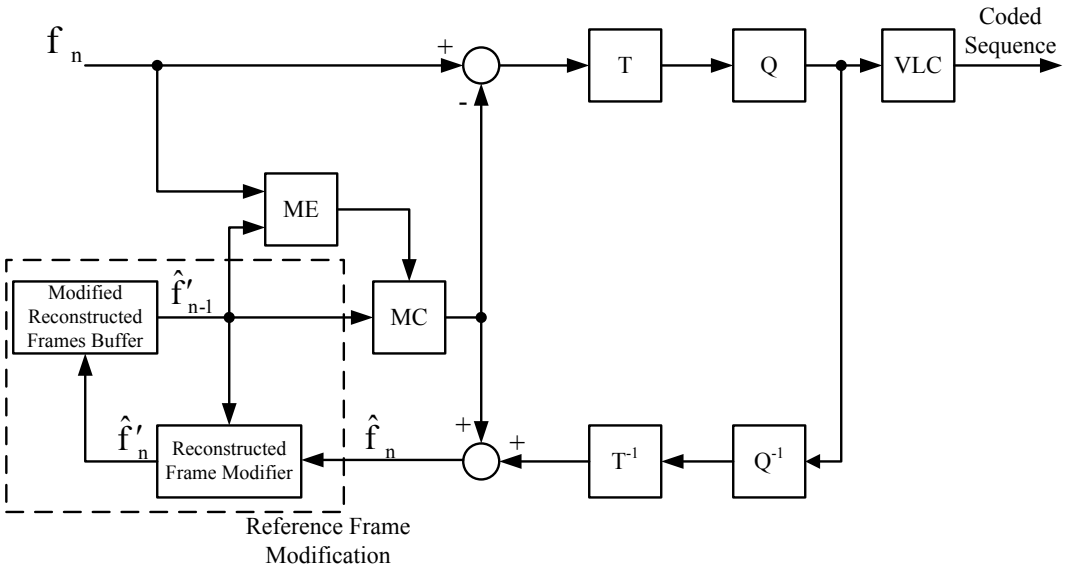


Fig. 2.8 Block diagram of a video encoder with reference frame modification. (ME, MC, T and Q represent Motion Estimation, Motion Compensation, Transform and Quantization respectively).

2.6 Reference Frame Modification Methods

In conventional video coding, each encoded frame is decoded and saved in a reconstructed frame buffer to be used as reference for predicting future frames. Due to the prediction structure, transmission errors may propagate through frames. One approach to lessen the effect of error propagation is to change the prediction structure by modifying the reconstructed frame. The modified reconstructed frame, which is usually less vulnerable to transmission errors, is utilized as the reference frame for motion compensation of future frames. These techniques are called reference frame modification (RFM) techniques. It should be noted that since the prediction structure of the standard is redesigned, these techniques are not standard compliant.

Fig. 2.8 and Fig. 2.9 respectively illustrate the block diagrams of a typical video encoder and decoder with reference frame modification components. It is assumed that simple picture copy is used for concealment of lost slices at the decoder. The reconstructed frame at the encoder and decoder are represented by $\hat{f}_n$ and $\tilde{f}_n$, and $\hat{f}'_n$ and $\tilde{f}'_n$ denote the modified reconstructed values at the encoder and decoder, respectively. The only part added comparing to conventional video coding is the Reconstructed Frame Modifier unit. It should be noted that $\hat{f}'_{n-1}$ and $\tilde{f}'_{n-1}$ are used as the reference frames in prediction of frame

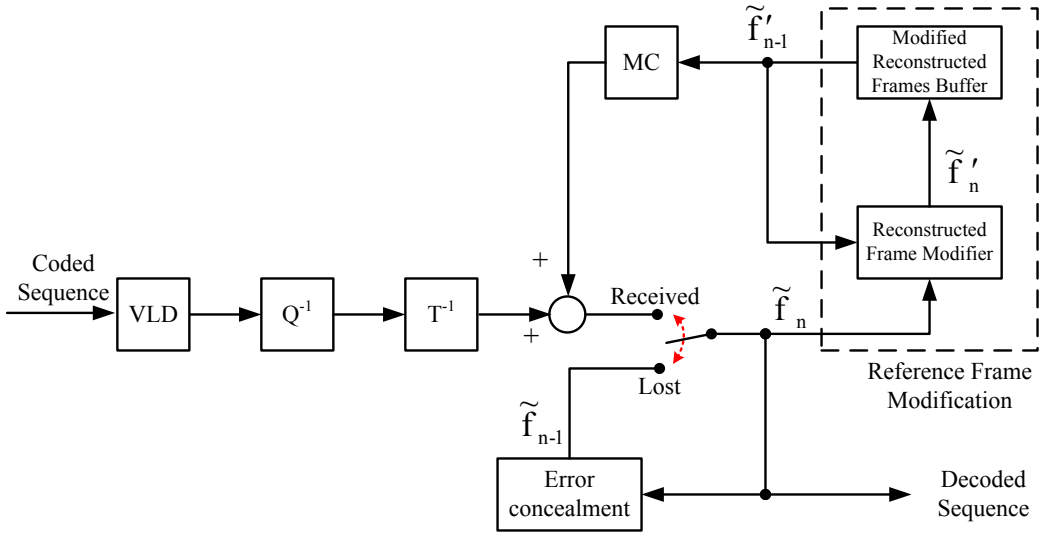


Fig. 2.9 Block diagram of a video decoder with reference frame modification.

n. In an error free case, their values are the same, which results in synchronized encoding and decoding prediction loops. In this section, some important reference modification methods are studied.

2.6.1 Leaky prediction

In leaky prediction, the modified reconstructed frame is the weighted sum of the reconstructed frame and a proper constant. This will result in exponential decay of the impact errors from previous frames [30]. Leaky prediction is defined as:

$$\hat{f'}_n = \alpha \hat{f}_n + (1-\alpha)K, \tag{2.25}$$

where $\hat{f}_n$ and $\hat{f'}_n$ denote the reconstructed frame and the modified one, and α and K are the leaky factor and a constant respectively. α may take a value between 0 and 1. It controls the trade off between coding efficiency and error resilience. Decreasing α results in a more resilient stream with lower prediction quality. Setting $\alpha = 1$ is equivalent to conventional video coding and when $\alpha = 0$ no temporal prediction is performed. α typically takes a value in the range of 0.8 – 0.95 [30, 33]. Analytical solutions for finding the near optimum alpha have been proposed [31,32] for layered coding. Fig. 2.10 shows the "Reference Frame

Modification" block of Fig. 2.8 for the leaky prediction.

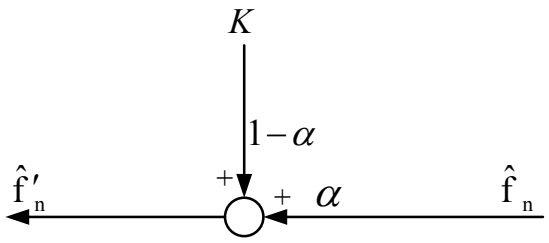


Fig. 2.10 The leaky prediction "Reference Frame Modification" block in Fig. 2.8.

K usually takes on the value of the mid range of pixel values which is 128 [115]. In SNR layered video coding methods, K can be chosen in a way that improves the coding efficiency. In these methods, the base layer is generally protected by using Forward Error Correction (FEC) and for enhancement layers K is replaced by the base layer reconstruction [116,117]:

$$\hat{f}'_{n,En} = \alpha \hat{f}_n + (1-\alpha)\hat{f}_{n,Base}, \tag{2.26}$$

where $\hat{f}_{n,Base}$ and $\hat{f}'_{n,En}$ denote the base layer reconstructed frame n and the enhancement layer modified reconstruction of frame n.

2.6.2 Generalized Source Channel Prediction (GSCP)

Generalized Source Channel Prediction (GSCP) [33] could be considered as an extension of leaky prediction which generates the modified reconstructed frame as a weighted sum of the current frame reconstruction and previous modified reconstruction frame. Fig. 2.11 shows the "Reference Frame Modification" block of Fig. 2.8 for the GSCP technique prediction. This modification is defined as:

$$\hat{f}'_n = \alpha \hat{f}_n + (1-\alpha)\hat{f}'_{n-1}. \tag{2.27}$$

Since the previous frame has more correlation with the current frame, employing $\hat{f}'_{n-1}$ instead of a constant (K) leads to a better prediction. In addition, propagating Intra coded MBs from previous frames to prediction of future frames improves the robustness. The near

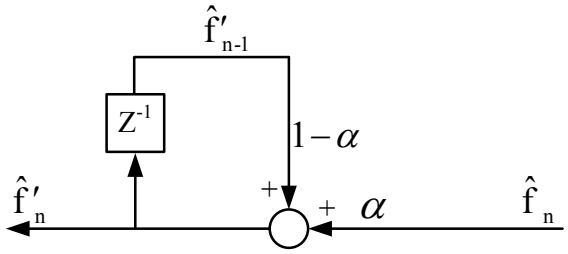


Fig. 2.11 The GSCP "Reference Frame Modification" block in Fig. 2.8.

optimal α is calculated as:

$$\alpha = 1 - p - H, \tag{2.28}$$

where p is the packet loss rate and H is a constant between 0.1 and 0.2 [33].

2.6.3 Improved Generalized Source Channel Prediction (IGSCP)

The Improved Generalized Source Channel Prediction (IGSCP) [34, 118] scheme improves GSCP by adding more emphasis on Intra MBs. Since Intra coded blocks do not propagate errors from previous pictures, the IGSCP technique copies the Intra coded blocks from the previous frame into the current modified reconstructed frame. Consequently, the new reconstructed frame, which is used as a reference for prediction of future frames, has more Intra coded blocks. Fig. 2.12 shows the "Reference Frame Modification" block of Fig. 2.8 for the IGSCP technique prediction. This technique is defined as:

$$\hat{f'}_n^i = \begin{cases} \hat{f}_{n-1}^i & \text{if } f_{n-1}^i \text{ is Intra coded,} \\ \alpha \hat{f}_n^i + (1-\alpha)\hat{f'}_{n-1}^i & \text{otherwise.} \end{cases} \quad \text{for } i = 1 \ldots I, \tag{2.29}$$

where f_n^i denotes the i^{th} pixel in the n^{th} frame. The number of pixels in a frame is represented by I. When the i^{th} pixel in the frame $n-1$ is inside an Intra coded MB, it will be copied to the modified reconstructed frame ($\hat{f'}_n^i$). In other cases, the GSCP technique is applied. Furthermore, an improvement of this technique, which requires a separate link between the encoder and the decoder, was proposed in [119]. The link is used for sending the internal decision made at the encoder to the decoder.

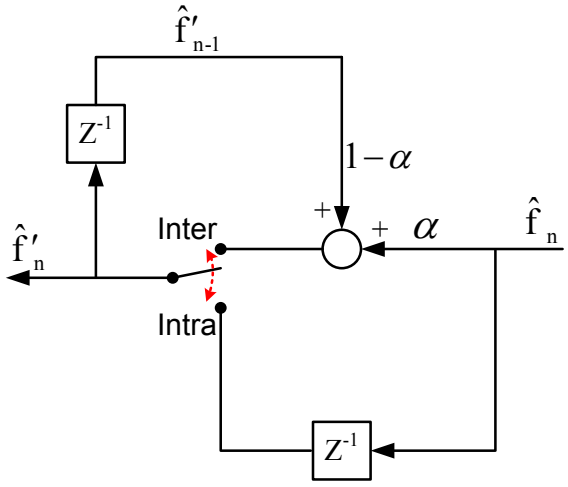


Fig. 2.12 The IGSCP "Reference Frame Modification" block in Fig. 2.8.

2.7 Chapter Summary

With rapid development of advanced multimedia applications, scalable video coding is widely used. In this chapter, we studied the scalable video coding and specifically the scalable extension of H.264/AVC known as SVC. By using SVC a flexible bitstream is produced that fulfils the requirements of clients with different temporal, spatial and quality scalability. The scalability is achieved at the cost of low degradation in coding efficiency in comparison to single layer coding. The base layer of SVC is compatible with H.264/AVC bitstream. The scalable extension of H.264/AVC exploits most features of H.264/AVC in addition to new features in temporal, spatial and quality Scalability.

Various techniques have been proposed to decrease the effect of transmission error on the decoded video quality. In this chapter, we studied some of the techniques that are used at the encoder. Intra updating is the most basic approach to solve the problem. Although inserting Intra MBs stops the error propagation, it consumes a lot of bits. Thus, the Intra MB should be effectively inserted in proper regions. These techniques were further improved in error resilience mode decision techniques. Instead of choosing between Intra and Inter modes, error resilience mode decision selects the best mode based on the source and channel states. In order to make an optimum mode decision, various methods estimate the end-to-end distortion. These methods were discussed and their short comings were explained.

Considering both channel and source distortion in choosing the best motion vectors is called error resilience motion estimation. This technique can be used in cooperation with error resilience mode decision. Another approach that can be employed independently of the above techniques is reference frame modification. In reference frame modification, the reference frame is modified to a new one which is less vulnerable to transmission error.

In the following chapters, we present novel techniques in order to decrease the quality degradation caused by the video transmission over error prone networks. In Chapter 3, we propose two reference frame modification techniques for temporal and spatial scalability that improve the previous methods by efficiently exploiting the Intra MBs in the reference frames and exponential decay of error propagation caused by the introduced leaky prediction. Also, we estimate the end-to-end distortion of the proposed prediction structure in order to be used in the rate distortion optimization. In Chapter 4, we present an accurate low complexity utility estimation technique that can be used in unequal protection of scalable bitstream.

Chapter 3

Reference Frame Modification Techniques

The conventional prediction structure can be modified in order to reduce the propagation of error to succeeding frames. In the conventional prediction structure, the current reconstructed frame is used as a reference for the motion estimation and the motion compensation of the following frames. In this new approach, the reconstructed frame is modified into a new one which is less vulnerable to transmission errors. The modified reconstructed frame is used as a reference in prediction of succeeding frames. These techniques are known collectively as reference frame modification (RFM) and are described in Section 2.6. Previously, our team has explored several error resilience techniques in different components of the encoder [18, 23, 24, 88, 119]. These techniques were examined in H.264/AVC. In this chapter, we propose two new reference frame modification techniques that can be used in temporal and spatial scalability of SVC. The first technique improves the previous RFM methods by efficiently using the Intra coded blocks in the previous frames. The second technique improves the leaky prediction structures by introducing a local spatial average value. Also, it makes use of previously Intra coded MBs in order to improve the error robustness. Furthermore, in order to get a better performance, RFM techniques can be combined with other error resilient methods such as random Intra refresh and error resilience mode decision. In order to combine our proposed techniques with error resilience mode decision, the end-to-end distortion of the second proposed technique is estimated.

We begin by extending the existing reference frame modification schemes to the tem-

poral and spatial scalability. We then proceed by introducing and explaining our proposed prediction structures and the reason they perform better than the existing schemes. We then focus on the end-to-end distortion calculation of our technique. Finally, we provide simulation results for all our proposed schemes showing improvements in performance.

3.1 Adaptation of Previous Methods in Scalable Video Coding

As it was mentioned in Chapter 2, SVC has been proposed as an extension of H.264/AVC and the base layer of a scalable encoded stream is completely compatible with H.264/AVC. Furthermore, most of the H.264/AVC components are also used in SVC. As a result, most of the error resilient methods in H.264/AVC can be adapted in SVC by performing some modification. The RFM techniques explained in Section 2.6 were applied for single layer coding. In this section, these methods are extended to work in temporal and spatial scalability.

In Fig. 3.1, the prediction structure of a scalable stream with four temporal and two spatial layers is shown. In the base temporal layer, frames are coded as I or P pictures, while in higher temporal layers, frames are bipredictive Inter pictures (B pictures). P pictures require only one reference picture list (*list0*) for prediction, while in B pictures two separate reference picture lists (*list0* and *list1*) are employed. Furthermore, in each spatial layer, in addition to independent temporal motion estimation and motion compensation, the base or other previous enhancement layers are used as a reference for inter-layer prediction. Fig. 3.1 also shows the dependencies between temporal and spatial layers caused by prediction structure. Dashed and solid arrows represent prediction from reference 0 and reference 1 respectively, and inter-layer prediction between spatial layers is denoted by vertical arrows. The numbers below each frame corresponds to the display order at the decoder. For instance, picture 9, which is a P picture in the base temporal layer, uses frame 1 as the only reference picture, while picture 5 utilizes frame 1 as reference 0 and picture 9 as reference frame 1. A base layer picture and the set of pictures between successive base layer pictures is referred to as Group of Pictures (GoP).

Based on the temporal prediction structure shown in Fig. 3.1, we divide frames into three groups. The first group contains frames in the base temporal layers. These pictures, which would be used as a reference for motion estimation and compensation of other frames, employ only one reference (f_{n_ref0}) in their temporal prediction. The second group includes

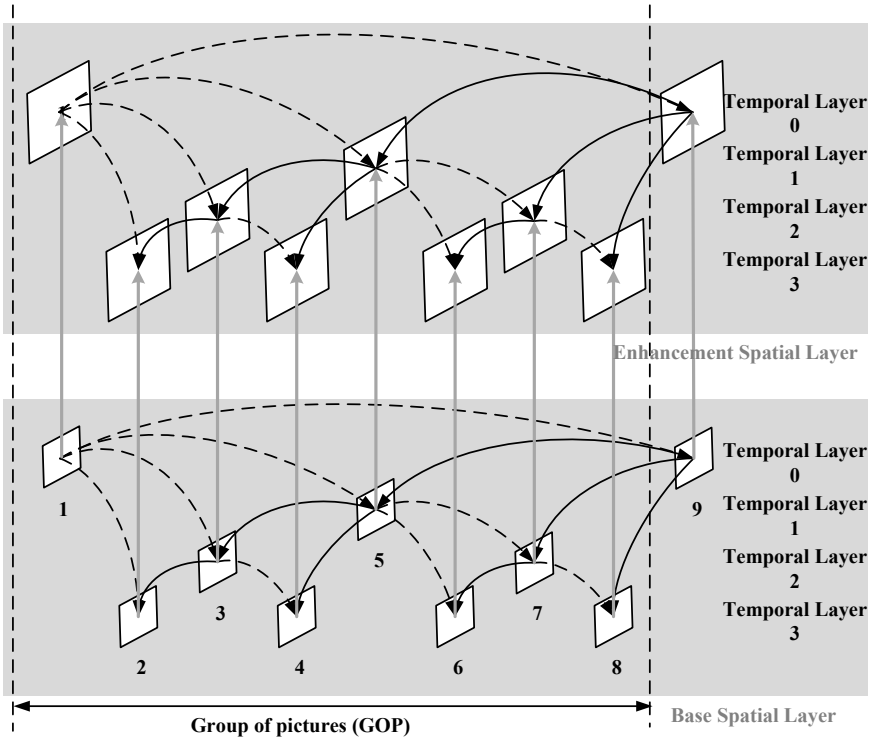


Fig. 3.1 Temporal and spatial scalable structure.

frames between the first and the last temporal layers. These frames make use of two references (f_{n_ref0} and f_{n_ref1}) for their temporal predictions. Finally, the frames in the highest temporal layer (frames 2, 4, 6 and 8 in this example) form the last group. These frames are not utilized as a reference for temporal prediction of other frames, and as a result, no reference frame modification is applied on them. Assuming a simple picture copy from the reference 0 as the error concealment of the decoder, the above RFM schemes can be easily extended to the temporal scalability. Leaky prediction can be employed without any changes. For GSCP and IGSCP techniques, it is required to replace $n-1$ by n_ref0. The modified reconstructed frame for GSCP can be stated as:

$$\hat{f}'_n = \alpha \hat{f}_n + (1-\alpha)\hat{f}'_{n\_ref0}, \tag{3.1}$$

to be compared with Eq. (2.27). For the IGSCP technique, the modified reconstructed frame is defined as:

$$\hat{f'}_n^i = \begin{cases} \hat{f}_{n\_ref0}^i & \text{if } f_{n\_ref0}^i \text{ is Intra coded,} \\ \alpha \hat{f}_n^i + (1-\alpha)\hat{f'}_{n\_ref0}^i & \text{otherwise.} \end{cases} \quad \text{for } i = 1 \dots I, \tag{3.2}$$

where index $_{n\_ref0}^{i}$ denotes the i^{th} pixel in the reference 0 of the n^{th} frame. Furthermore, spatial scalability can be combined with temporal scalability. In this case, a similar scheme as mentioned above can be used for the base spatial layer. The modified reconstructed frame in higher spatial layers can be formed by using information from the base or previous spatial layers. In this way, the reconstructed frame of the base spatial layer is upsampled ($\hat{f}_{n\_UpsBase}$) to match the size of the enhancement layer and is used instead of the reference 0 of n^{th} frame. The GSCP technique for enhancement spatial layers is defined as:

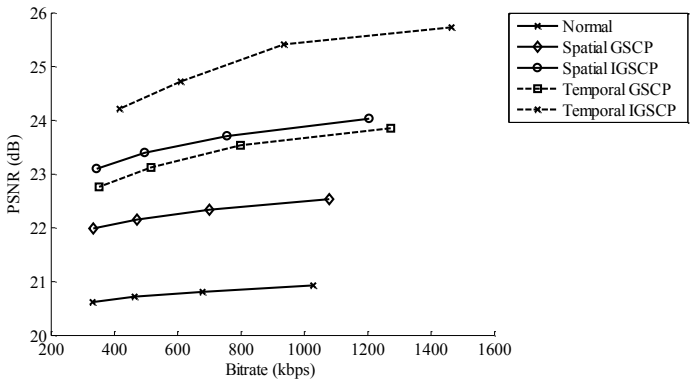
$$\hat{f'}_n = \alpha \hat{f}_n + (1-\alpha)\hat{f'}_{n\_UpsBase}. \tag{3.3}$$

For IGSCP, we have:

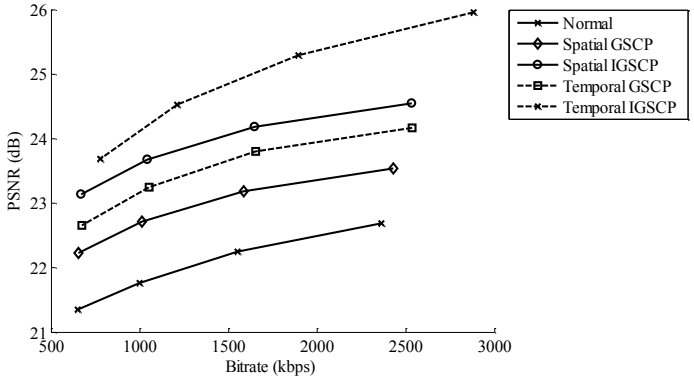
$$\hat{f'}_n^i = \begin{cases} \hat{f}_{n\_UpsBase}^i & \text{if } f_{n\_UpsBase}^i \text{ is Intra coded,} \\ \alpha \hat{f}_n^i + (1-\alpha)\hat{f'}_{n\_UpsBase}^i & \text{otherwise.} \end{cases} \quad \text{for } i = 1 \dots I. \tag{3.4}$$

In order to make it unambiguous in the rest of the text, we refer to Eq. (3.1) and Eq. (3.2) as temporal GSCP and temporal IGSCP respectively, and refer to Eq. (3.3) and Eq. (3.4), which use the base spatial layer in forming the modified reconstructed frame, as spatial GSCP and spatial IGSCP.

We observed that applying temporal GSCP and IGSCP achieve better performance compared to the spatial GSCP and IGSCP. The improvement was more significant in sequences with slow movement or high spatial details . In slow movement sequences, successive frames have more correlation, so using temporal references is a better choice. In high detail sequences, since upsampling the base layer does not show all the details, using temporal references usually achieves better results. Fig. 3.2 shows the rate distortion curves of each method for the "Foreman" and "Football" standard sequences with two spatial layers, five temporal layers, and frame rate of 30 fps. The packet loss rate is set to 10%, and 15% of MBs are randomly encoded as Intra.



(a) Foreman



(b) Football

Fig. 3.2 Rate distortion curves for different methods with two spatial and five temporal layers (a) "Foreman" sequence and (b) "Football" sequence with packet loss rate of 10% and 15% Intra refreshing.

3.2 Proposed Prediction Structures

3.2.1 The first proposed structure

Due to the fact that Intra MBs effectively restrain the propagation of errors in hybrid video coding, we improve IGSCP by putting more emphasis on Intra coded MBs. The proposed method makes use of both reference frames, upon availability, to improve the robustness. This modification is performed in three ways. First, assuming T as the number of temporal layers and considering the structure in Fig. 3.1, the pictures in the last temporal layer $(T-1)$ are not used as reference pictures for future prediction. Therefore, there is no need to modify the reconstructed picture, and so no RFM is used on those pictures. For the base temporal

layer, the situation is similar to the IPPP case, where each frame uses only its previous frame as its reference for prediction, thus IGSCP is applied. For other temporal layers, the modes of co-located MBs in each reference frame are checked. If the mode of one of the references is Intra, the Intra coded MB in the reference frame is copied to the modified reconstructed frame. If the co-located MBs in both references are Intra coded, the average is taken and used as the modified reconstruction value of the current macroblock. In other cases such that none of the co-located MBs in reference frames are Intra coded, the weighted sum of the current frame reconstructed macroblock and the modified reconstruction of co-located MB in reference 0 is used as the current frame modified reconstruction MB, as in GSCP. The reason that we only consider reference 0 in the last case is because of the default error concealment method used at the decoder that covers the lost slice by copying from reference 0. If the concealment method utilizes reference frame 1 instead of reference frame 0, using reference frame 1 instead of reference frame 0 for GSCP leads to better results.

IGSCP and the above method provide satisfactory performance by exploiting Intra MBs in reference frames for formation of modified reconstructed frames. Although copying Intra MBs from reference frames improves the robustness, it decreases the coding efficiency. The coding efficiency reduction is more noticeable in hierarchical structures in which the temporal distance between the current frame and the reference frames might be more than one. In addition, in low packet loss rates, copying the Intra MBs from the reference frame results in lower performance compared to GSCP. In order to address this problem, we introduce new coefficients for each reference frame. When the co-located macroblock in a reference frame is Intra coded, instead of copying the exact values of the pixels, the weighted sum of Intra coded values in the reference frames and current reconstructed frames is calculated. Finally, the proposed method is given by:

If temporal layer $= 0$:

$$\hat{f'}^{i}_{n} = \begin{cases} w_0 \hat{f}^{i}_{n\_ref0} + (1 - w_0)\hat{f}^{i}_{n} & \text{if } f^{i}_{n\_ref0} \text{ is Intra coded,} \\ \alpha \hat{f}^{i}_{n} + (1 - \alpha)\hat{f'}^{i}_{n\_ref0} & \text{otherwise.} \end{cases} \tag{3.5}$$

Else If $0 <$ temporal layer $< T - 1$:

$$\hat{f'}_n^i = \begin{cases} \frac{w_0\hat{f}^i_{n\_ref0}+w_1\hat{f}^i_{n\_ref1}+(2-w_0-w_1)\hat{f}^i_n}{2} & \text{if } f^i_{n\_ref0} \text{ \& } f^i_{n\_ref1} \text{ are Intra coded,} \\ w_0\hat{f}^i_{n\_ref0}+(1-w_0)\hat{f}^i_n & \text{else if } f^i_{n\_ref0} \text{ is Intra coded,} \\ w_1\hat{f}^i_{n\_ref1}+(1-w_1)\hat{f}^i_n & \text{else if } f^i_{n\_ref1} \text{ is Intra coded,} \\ \alpha\hat{f}^i_n+(1-\alpha)\hat{f'}^i_{n\_ref0} & \text{otherwise.} \end{cases} \tag{3.6}$$

$$for\ i = 1 \dots I,$$

Else If temporal layer $= T-1$:

$$\hat{f'}_n = \hat{f}_n. \tag{3.7}$$

As before, index ^i_n refers to the i^{th} pixel in the n^{th} frame, and $\hat{f}$ and $\hat{f'}$ denote the reconstructed and modified reconstructed frame respectively. Furthermore, n_ref0 and n_ref1 denote the reference 0 and reference 1 of frame n. T and I are the number of temporal layers and the number of pixels in one frame. α is the leaky factor introduced before and the near optimal α is calculated as (2.28). w_0 and w_1 are the weights of reference 0 and reference 1. The block diagram of the formation of the new reference in this method is shown in Fig. 3.3.

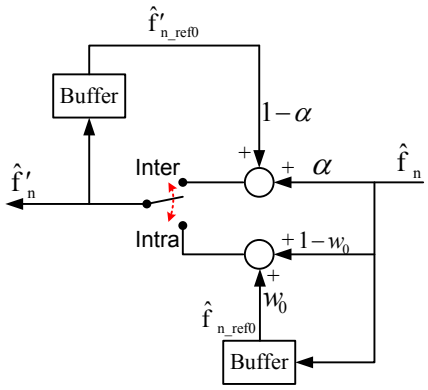
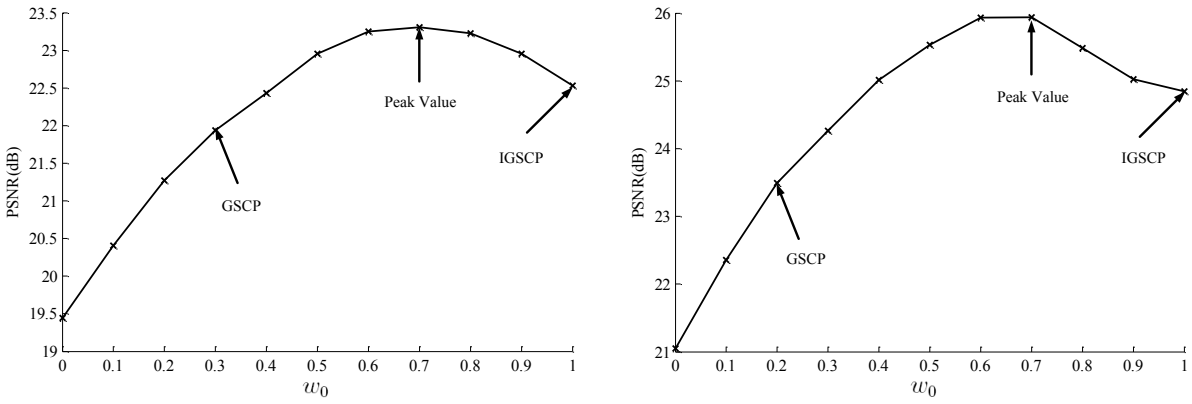


Fig. 3.3 The first proposed prediction structure block diagram.

Decreasing w_0 and w_1 to $1-\alpha$ results in the GSCP method and increasing them to one is equivalent to the IGSCP, but considering both reference frames. Fig. 3.4 shows the effect



(a) Bus, PLR = 20%

(b) Paris, PLR = 10%

Fig. 3.4 Average PSNR vs. w_0 for proposed method for (a) "Bus" at packet loss rate of 20%, and (b) "Paris" at packet loss rate of 10%. 15 fps at 500 kbps and 15% Intra refreshing.

of changing w_0 and w_1 values in the received video quality and assuming $w_0 = w_1$. Based on Eq. (2.28), near optimum values of α for packet loss rates of 10% and 20% are equal to 0.8 and 0.7, respectively. Therefore, using $w_0 = 0.3$ in Fig. 3.4-a and $w_0 = 0.2$ in Fig. 3.4-b convert this method to GSCP. It can be observed that increasing these coefficients to 1 does not improve the average PSNR monotonically and there is a peak value in each curve. Finding the optimum coefficients requires exhaustive simulations which are not always applicable. Our simulation results show that the near optimum weights are usually between 0.7 and 0.6. For error free cases, there is no need to copy Intra coded pixels from reference frames, so these values are set to 0.

3.2.2 The second proposed structure

As mentioned before, leaky prediction will result in exponential decay of error propagation in previous frames. However, using a constant value (K) in prediction significantly decreases the coding efficiency, especially in slow sequences such as "Akiyo" and "News". In slow sequences, the majority of the MBs are encoded by predictive coding, and the role of motion estimation is more noticeable. Since using leaky prediction reduces the prediction effectiveness, it will result in quality degradation more considerably in slow sequences compared to medium and fast motion sequences. In this method, our goal is to make use of the advantage of leaky prediction in mitigating the effect of transmission error, while

ensuring that the coding efficiency is sustained.

K is usually set to 128 which is the mid range of pixel values (0-255). However, it was observed through simulations that 128 is not the best choice for all sequences. In each sequence, based on the content of the video, the range of pixel values varies. A better choice is to take the average of pixel values for all frames and use it as the constant. In this way, K takes a different value for each sequence. Although this technique achieves good performance, it requires to transmit the constant to the decoder which is not applicable in all cases.

In order to improve this technique further, we introduced a new leaky value ($\hat{K}_n^i$). This new value, which is the average of reconstructed pixel values of neighbouring MBs, has more correlation with current MB and will result in better prediction. Since all the pixels in each block would have the same leaky value, the leaky value is calculated once for each block and used for all the pixels inside the block. As shown in Fig. 3.5, each MB may have a different number of neighbours based on its location in the frame. The bold square shows the current MB and shaded ones represent the neighbouring MBs. The average is taken over the available neighbours. It should be mentioned that the formation of a new reconstructed frame is done after encoding all MBs in the current frame. As a result, all the shaded MBs are available for calculation of the new leaky value. Furthermore, it is not required to transmit the new constant. At the receiver, the decoder does the same in calculating the value of $\tilde{K}_n^i$. For the cases when the neighbouring MBs are lost, the decoder first applies the concealment method, and then calculates the local average. Since the average is typically taken over more than 2000 pixels, the difference between $\hat{K}_n^i$ and $\tilde{K}_n^i$ is usually negligible. Table 3.1 shows the average $\hat{K}_n^i$ and the average and standard deviation of the absolute difference ($|\hat{K}_n^i - \tilde{K}_n^i|$) at the encoder and the decoder for different sequences. All the videos are encoded at 512 kbps and transmitted over channels with 10% packet loss. The low values reported in Table 3.1 validate our reasoning.

Utilizing the local spatial average ($\hat{K}_n^i$) leads to more efficient temporal prediction especially in sequences with slow movement. Fig. 3.6 illustrates the effect of using different constants in the leaky prediction technique. Two slow sequences are selected in which the improvements are more considerable. Using the new local average results in a gain of up to 2 dB for "Akiyo" sequence compared to normal leaky prediction. However, it was noticed that for sequences with medium or fast movements the improvement is lower, or even in some cases, there is no improvement. But since the problem of quality degradation caused

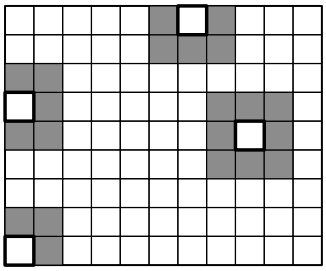


Fig. 3.5 Different positions of a MB and its neighbouring MBs in a frame.

Table 3.1 Average $\hat{K}_n^i$, average difference and standard deviation between $\hat{K}_n^i$ and $\tilde{K}_n^i$ for different sequences with CIF size at 10% packet loss rate and bit rate of 512 kbps.

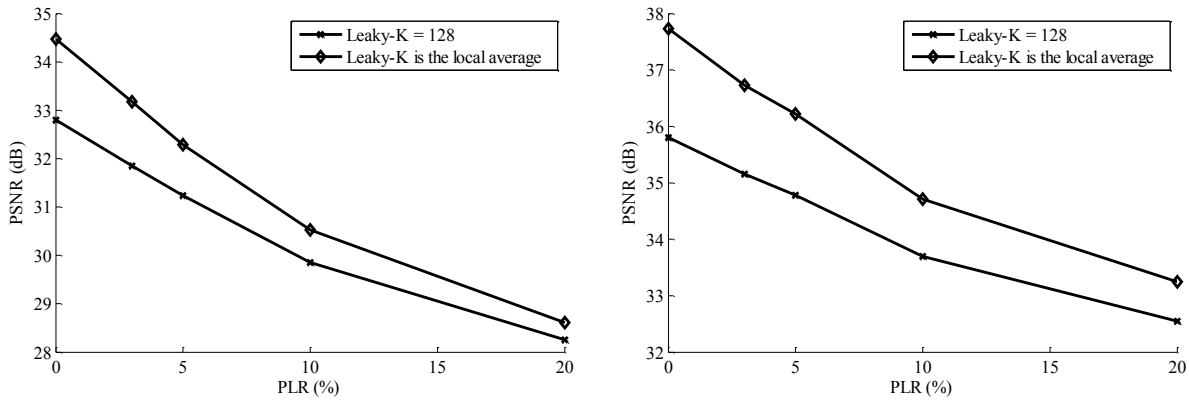
Sequence	Average $\hat{K}_n^i$	Average $\lvert\hat{K}_n^i - \tilde{K}_n^i\rvert$	Standard Deviation $\lvert\hat{K}_n^i - \tilde{K}_n^i\rvert$
Akiyo	46.76	0.83	0.10
Bus	41.11	0.90	0.35
Flower	98.86	0.98	0.41
Football	53.81	0.44	0.24
Foreman	80.08	0.73	0.17
Mobile	66.14	0.72	0.10
News	39.24	0.59	0.11
Paris	53.24	0.77	0.25
Stefan	66.87	0.53	0.14

by the leaky prediction is more significant in slow sequences, this technique helps improving the quality of the decoded video in general.

The performance of this method is further improved by combining it with the IGSCP technique. In order to use Intra coded blocks more efficiently, both temporal reference pictures are considered upon availability. Based on the hierarchical prediction structure used in temporal scalability, pictures are divided into three categories:

1. The first group contains frames in the base temporal layer. These frames make use of one reference picture for temporal prediction. These frames have the IPPP structure which is used in H.264/AVC. Pixels in any co-located Intra MB in the reference frame (f_{n_ref0}) are copied to the modified reconstructed frame.

2. The pictures between the first and highest temporal layers make up the second group.



(a) News. (b) Akiyo.

Fig. 3.6 PSNR vs. Packet loss rate for leaky prediction with different constant values for (a) "News" with QCIF size and 15 fps at 128 kbps and (b) "Akiyo" with QCIF size and 15 fps at 128 kbps and 15% Intra refreshing.

All these frames use two reference pictures for prediction. If the co-located MBs in either of the reference frames (f_{n_ref0} or f_{n_ref1}) are Intra coded, the pixels will be copied to the current modified reconstructed frame ($\hat{f}'^{i}_{n}$). If both are Intra coded, the average is taken and used as the modified reconstruction value of pixel i.

3. The last group includes pictures in the highest temporal layer (pictures 2, 4, 6 and 8). No reference frame modification is applied on these frames. The reason is that these frames are not utilized as reference pictures for future prediction.

In the first and second groups, if the co-located MBs in reference frames are Intra coded, a weighted summation of the current reconstructed frame ($\hat{f}^{i}_{n}$), the previous reference frame ($\hat{f}'^{i}_{n\_ref0}$), and the local spatial average ($\hat{K}^{i}_{n}$) is computed and used as the modified reconstructed frame of the i^{th} pixel. The proposed technique uses (i) error robustness of previously Intra coded pixels, (ii) exponential decay of error propagation caused by leaky prediction, and (iii) good prediction resulting from using the new local spatial average simultaneously. This method is defined as:

If temporal layer = 0:

$$\hat{f}'^{i}_{n} = \begin{cases} \hat{f}^{i}_{n\_ref0} & \text{if } f^{i}_{n\_ref0} \text{ is Intra coded,} \\ (\alpha-\beta)\hat{f}^{i}_{n} + (1-\alpha)\hat{f}'^{i}_{n\_ref0} + \beta\ \hat{K}^{i}_{n} & \text{otherwise.} \end{cases} \tag{3.8}$$

$$\text{for } i = 1 \dots I,$$

Else If $0 <$ temporal layer $< T-1$:

$$\hat{f'}_n^i = \begin{cases} \frac{\hat{f}^i_{n\_ref0} + \hat{f}^i_{n\_ref1}}{2} & \text{if } f^i_{n\_ref0} \text{ \& } f^i_{n\_ref1} \text{ are Intra coded,} \\ \hat{f}^i_{n\_ref0} & \text{else if } f^i_{n\_ref0} \text{ is Intra coded,} \\ \hat{f}^i_{n\_ref1} & \text{else if } f^i_{n\_ref1} \text{ is Intra coded,} \\ (\alpha-\beta)\hat{f}^i_n + (1-\alpha)\hat{f'}^i_{n\_ref0} \; + \; \beta \; \hat{K}^i_n & \text{otherwise.} \end{cases} \tag{3.9}$$

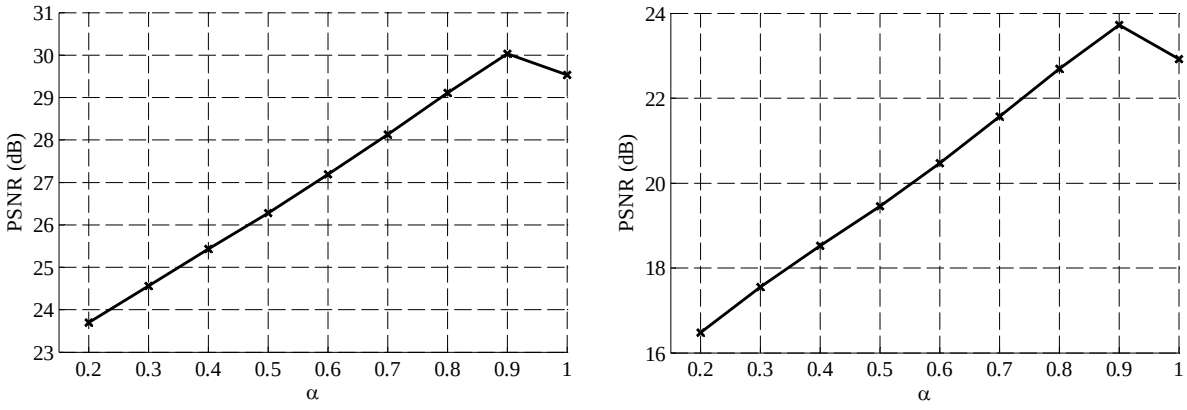
$$\text{for } i = 1 \dots I,$$

Else If temporal layer $= T-1$:

$$\hat{f'}_n = \hat{f}_n. \tag{3.10}$$

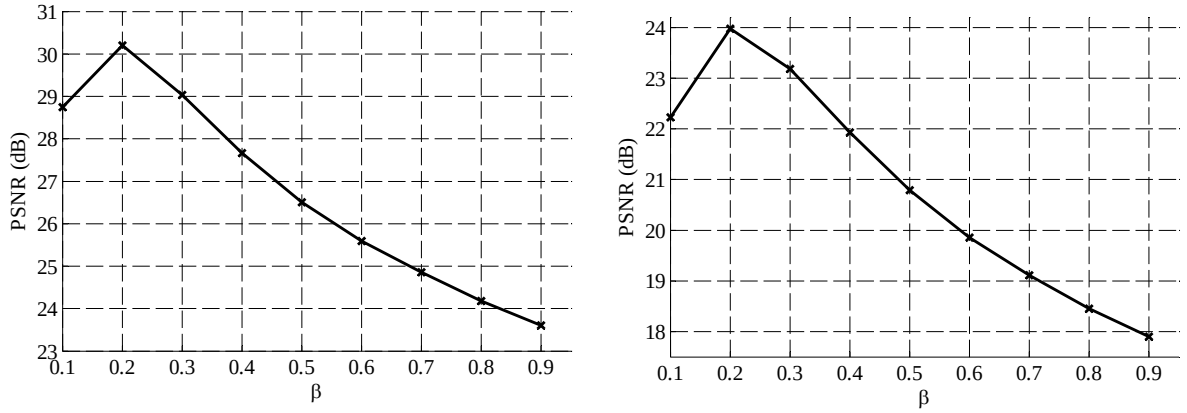
where T shows the number of temporal layers and I is the number of pixels in each frame. ^i_n denote the i^{th} pixel in the n^{th} frame, and $\hat{f}$ and $\hat{f}'$ represent the reconstructed and modified reconstructed frame respectively. $\hat{K}^i_n$ is calculated as the average of pixel values in the neighbouring MBs. It is calculated once for all the pixels inside a MB. β denotes the weight of $\hat{K}^i_n$ and can take on a value between 0 and α. α represents the leaky factor introduced before. Lower prediction efficiency or better error robustness is achieved by increasing β. On the other hand, decreasing it results in lower error robustness or better prediction. Several simulations for different types of sequences were conducted in order to tackle this trade off. The simulation results show that near optimum β is around 0.2; however, for slow sequences smaller values (0.05) lead to better performance. Fig. 3.7 and Fig. 3.8 show the impact of changing the value of α and β for "Foreman" and "Mobile" sequences at packet loss rate of 5%, respectively. The block diagram of the formation of the new reference in this method is shown in Fig. 3.9.

Furthermore, in order to improve the error robustness in spatial enhancement layer, the pixels in the Intra MBs in the lower spatial layer are upsampled and copied to the current frame modified reconstruction in the same way as spatial IGSCP (Eq. (3.4)). If the co-located MB is also Intra coded in any of the references, the average of Intra pixels in the lower spatial layer and reference frames are used instead. It should be noted, in calculating



(a) Foreman (b) Mobile

Fig. 3.7 Average PSNR vs. α for the proposed method for (a) "Foreman", and (b) "Mobile" at packet loss rate of 5%. Encoded at 2048 kbps and 10% Intra refreshing.



(a) Foreman (b) Mobile

Fig. 3.8 Average PSNR vs. β for the proposed method for (a) "Foreman", and (b) "Mobile" at packet loss rate of 5%. Encoded at 2048 kbps and 10% Intra refreshing.

the weighted sum, only the reference frame 0 is considered. It is because of the assumption that the error concealment method conceals the lost slice by copying from the reference 0. It should be noted that the modified reconstructed frame values are rounded, and similar to reconstructed frame, are represented in integer values. Consequently, motion estimation and motion compensation are done as usual on integer values..

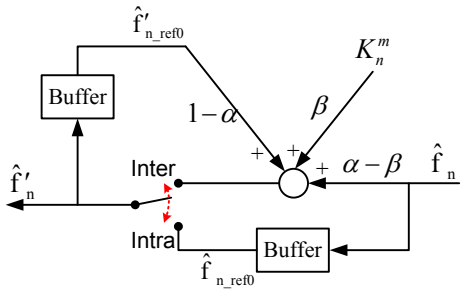


Fig. 3.9 The second proposed prediction structure block diagram.

3.3 End-to-End Distortion Estimation for the Proposed Scheme

In order to improve the performance, reference frame modification techniques can be used in combination with error resilient mode decision methods, described in Section 2.4. Error resilient mode decision methods usually select the best mode by using the estimated end-to-end distortion in mode decision process. However, using RFM techniques changes the prediction structure used in the encoding and decoding. As a result, the end-to-end distortion estimation will change based on the employed reference frame modification technique, and to the best of our knowledge, this issue has not been addressed before. In order to study the performance of our proposed techniques in co-operation with error resilience mode decision techniques, we modified the end-to-end distortion estimation of LARDO [28] (Section 2.5.4) based on the prediction structure used in our second proposed prediction structure.

In RFM methods, if the i^{th} pixel in frame n is Inter coded, the reconstructed pixel in the frames at the encoder will be:

$$\hat{f}_n^i = \hat{f}'^j_{ref} + \hat{r}_n^i. \tag{3.11}$$

It is assumed that the reference of i^{th} pixel in the n^{th} frame is pixel j in the ref^{th} frame (typically $ref = n-1$). The quantized prediction error is denoted by $\hat{r}_n^i$. At the decoder side, when a pixel is Inter coded, the reconstructed frame will be:

$$\tilde{f}_n^i = \begin{cases} \tilde{f}'^j_{ref} + \hat{r}_n^i & w.p. \quad 1-p, \\ \tilde{f}_{n-1}^i & w.p. \quad p. \end{cases} \tag{3.12}$$

where p is the probability of packet loss. It is assumed that simple picture copy is used as the error concealment method at the decoder. In other words, if pixel i is lost, it will be concealed by copying pixel i from frame $n-1$. For an Intra coded pixel, we will have:

$$\tilde{f}_n^i = \begin{cases} \hat{f}_n^i & w.p. \quad 1-p, \\ \tilde{f}_{n-1}^i & w.p. \quad p. \end{cases} \tag{3.13}$$

Assuming the reconstructed pixel at the decoder ($\tilde{f}_n^i$) is a random variable, the end-to-end distortion for Inter coded pixel is calculated by considering whether or not $\tilde{f}_n^i$ is lost, as:

$$\begin{aligned} d(n,i) &= E\Big\{\big(f_n^i - \tilde{f}_n^i\big)^2\Big\} \\ &= (1-p)E\Big\{\Big(f_n^i - (\tilde{f'}_{ref}^{\,j} + \hat{r}_n^i)\Big)^2\Big\} + pE\Big\{\big(f_n^i - \tilde{f}_{n-1}^i\big)^2\Big\} \\ &= (1-p)E\Big\{\Big(f_n^i - (\tilde{f'}_{ref}^{\,j} + \hat{f}_n^i - \hat{f'}_{ref}^{\,j})\Big)^2\Big\} + pE\Big\{\big(f_n^i - \tilde{f}_{n-1}^i\big)^2\Big\} \\ &= (1-p)\Big(E\Big\{\big(f_n^i - \hat{f}_n^i\big)^2\Big\} + E\Big\{\big(\hat{f'}_{ref}^{\,j} - \tilde{f'}_{ref}^{\,j}\big)^2\Big\}\Big) + pE\Big\{\big(f_n^i - \tilde{f}_{n-1}^i\big)^2\Big\} \quad (3.14) \\ &= (1-p)\Big(d_{src}(n,i) + d'_{ep}(ref,j)\Big) + pd_{ec}(n,i), \quad (3.15) \end{aligned}$$

where $\hat{f}_n^i$ and $\tilde{f}_n^i$ denote the reconstructed value of pixel i at the encoder and at the decoder, respectively. The modified reconstructed value at the encoder and decoder are represented by $\hat{f'}_n^{\,i}$ and $\tilde{f'}_n^{\,i}$. Eq. (3.14) is based on the assumption that effects of source distortion at the encoder and error propagation at the decoder are additive. Source and error concealment distortions are represented by $d_{src}(n,i)$ and $d_{ec}(n,i)$ respectively. $d'_{ep}(n,i)$ is the error propagation distortion in modified reconstructed frame and is calculated differently depending on the RFM method used. It should be noted that when the i^{th} pixel in the n^{th} frame is Intra coded, there is no error propagation from previous frames. Therefore, the end-to-end distortion is calculated as:

$$\begin{aligned} d(n,i) &= (1-p)E\Big\{\big(f_n^i - \hat{f}_n^i\big)^2\Big\} + pE\Big\{\big(f_n^i - \tilde{f}_{n-1}^i\big)^2\Big\} \\ &= (1-p)d_{src}(n,i) + pd_{ec}(n,i). \end{aligned} \tag{3.16}$$

Using the estimated distortion, Eq. (2.2) is modified to:

$$\begin{aligned} J_{\text{mode}} &= D + \lambda'_{\text{mode}} R_{\text{mode}} \\ &= (1-p)\Big(D_{src} + D'_{ep}\Big) + pD_{ec} + \lambda'_{\text{mode}} R_{\text{mode}}, \end{aligned}$$

where D is the sum of Eq. (3.15) or Eq. (3.16) over all pixels of the MB, depending on whether the MB is Inter or Intra coded. Since the error concealment is independent of the selected coding mode, there is no need to calculate D_{ec} for the optimization. Furthermore, based on [28], $\lambda'_{\text{mode}} = (1-p)\lambda_{\text{mode}}$. So, we obtain equivalently:

$$J'_{\text{mode}} = D_{src} + D'_{ep} + \lambda_{\text{mode}} R_{\text{mode}}. \tag{3.17}$$

Since source distortion can easily be calculated during encoding, the main issue will be the calculation of the error propagation distortion. This distortion is calculated recursively and stored for future use. In the proposed RFM technique, the modified reconstructed frame at the decoder ($\tilde{f'}_n^i$) is formed based on the mode of the co-located MB in the previous frame and the loss probabilities of current and previous frames. In Table 3.2, the different possible values of $\tilde{f'}_n^i$ are listed. It should be mentioned that when one block is lost, the decoder will not have any information about the mode of that block. Since the number of Intra MBs is usually less than the number of Inter coded MBs, the decoder assumes that the lost MB was coded as Inter. In the following, $d'_{ep}(n,i)$ is calculated for each of the four different cases defined in Table 3.2.

Table 3.2 Different values of current modified reconstructed frame at the decoder ($\tilde{f'}_n^i$).

f_{n-1}^i	$\tilde{f}_n^i$	$\tilde{f}_{n-1}^i$	$\tilde{f'}_n^i$	Case #	Prob.
Inter	not Lost	not Lost	$(\alpha-\beta)\tilde{f}_n^i + (1-\alpha)\tilde{f'}_{n-1}^i + \beta\tilde{K}_n^i$	I	$(1-p)^2$
	not Lost	Lost	$(\alpha-\beta)\tilde{f}_n^i + (1-\alpha)\tilde{f'}_{n-1}^i + \beta\tilde{K}_n^i$	I	$(1-p)p$
	Lost	not Lost	$\tilde{f'}_{n-1}^i$	II	$(1-p)p$
	Lost	Lost	$\tilde{f'}_{n-1}^i$	II	p^2
Intra	not Lost	not Lost	$\tilde{f}_{n-1}^i$	III	$(1-p)^2$
	not Lost	Lost	$(\alpha-\beta)\tilde{f}_n^i + (1-\alpha)\tilde{f'}_{n-1}^i + \beta\tilde{K}_n^i$	IV	$(1-p)p$
	Lost	not Lost	$\tilde{f}_{n-1}^i$	III	$(1-p)p$
	Lost	Lost	$\tilde{f'}_{n-1}^i$	II	p^2

CASE I: In this case, the co-located MB in the previous frame was coded as Inter and the current block was received correctly. So, $\tilde{f'}^{i}_{n}$ is formed as the weighted summation of current reconstructed frame ($\tilde{f}^{i}_{n}$), previous modified reconstructed one ($\tilde{f'}^{i}_{n-1}$) and the leaky factor ($\tilde{K}^{i}_{n}$).

$$
\begin{aligned}
d'_{ep\_I}(n,i) &= E\left\{\left(\hat{f'}^{i}_{n}-\tilde{f'}^{i}_{n}\right)^2\right\} \\
&= E\left\{\left((\alpha-\beta)\hat{f}^{i}_{n}+(1-\alpha)\hat{f'}^{i}_{n-1}+\beta\hat{K}^{i}_{n}-\left((\alpha-\beta)\tilde{f}^{i}_{n}+(1-\alpha)\tilde{f'}^{i}_{n-1}+\beta\tilde{K}^{i}_{n}\right)\right)^2\right\} \\
&= E\left\{\left((\alpha-\beta)(\hat{f}^{i}_{n}-\tilde{f}^{i}_{n})+(1-\alpha)(\hat{f'}^{i}_{n-1}-\tilde{f'}^{i}_{n-1})+\beta(\hat{K}^{i}_{n}-\tilde{K}^{i}_{n})\right)^2\right\} \\
&= E\left\{\left((\alpha-\beta)(\hat{f'}^{j}_{ref}+\hat{r}^{i}_{n}-\tilde{f'}^{j}_{ref}-\hat{r}^{i}_{n})+(1-\alpha)(\hat{f'}^{i}_{n-1}-\tilde{f'}^{i}_{n-1})\right)^2\right\} \qquad (3.18) \\
&= (\alpha-\beta)^2 E\left\{\left(\hat{f'}^{j}_{ref}-\tilde{f'}^{j}_{ref}\right)^2\right\}+(1-\alpha)^2 E\left\{\left(\hat{f'}^{i}_{n-1}-\tilde{f'}^{i}_{n-1}\right)^2\right\} \\
&\qquad +2(\alpha-\beta)(1-\alpha)E\left\{\hat{f'}^{j}_{ref}-\tilde{f'}^{j}_{ref}\right\}E\left\{\hat{f'}^{i}_{n-1}-\tilde{f'}^{i}_{n-1}\right\} \qquad (3.19) \\
&= (\alpha-\beta)^2 d'_{ep}(ref,j)+(1-\alpha)^2 d'_{ep}(n-1,i) \\
&\qquad +2(\alpha-\beta)(1-\alpha)dif_{ep}(ref,j)dif_{ep}(n-1,i). \qquad (3.20)
\end{aligned}
$$

Eq. (3.18) is based on assumption that the difference between the average $\hat{K}^{i}_{n}$ and $\tilde{K}^{i}_{n}$ is negligible. This assumption is based on Table 3.1. Also, Eq. (3.19) is based on the assumption that the mean modified reconstructed frame error at pixel i and j are independent. In the case that $i=j$ and $ref=n-1$, Eq. (3.19) is easily modified to $d'_{ep}(n-1,i)$. $dif_{ep}(n,i)$ is calculated for each of the four different cases defined in Table 3.2. For this case and with similar calculation, we obtain:

$$
dif_{ep\_I}(n,i)=E\left\{\hat{f'}^{i}_{n}-\tilde{f'}^{i}_{n}\right\}=(\alpha-\beta)dif_{ep}(ref,j)+(1-\alpha)dif_{ep}(n-1,i). \qquad (3.21)
$$

CASE II: In this case, the current frame is lost and the decoder knows that the co-located pixel in previous frame was either lost or Inter coded. As a result, the decoder does the concealment by copying the previous modified reconstructed frame.

$$
\begin{aligned}
d'_{ep\_II}(n,i) &= E\left\{\left(\hat{f'}^{i}_{n} - \tilde{f'}^{i}_{n-1}\right)^2\right\} \\
&= E\left\{\left((\hat{f'}^{i}_{n} - \hat{f'}^{i}_{n-1}) + (\hat{f'}^{i}_{n-1} - \tilde{f'}^{i}_{n-1})\right)^2\right\} \\
&= (\hat{f'}^{i}_{n} - \hat{f'}^{i}_{n-1})^2 + E\left\{(\hat{f'}^{i}_{n-1} - \tilde{f'}^{i}_{n-1})^2\right\} + 2(\hat{f'}^{i}_{n} - \hat{f'}^{i}_{n-1})E\left\{\hat{f'}^{i}_{n-1} - \tilde{f'}^{i}_{n-1}\right\} \\
&= (\hat{f'}^{i}_{n} - \hat{f'}^{i}_{n-1})^2 + d'_{ep}(n-1,i) + 2(\hat{f'}^{i}_{n} - \hat{f'}^{i}_{n-1})dif_{ep}(n-1,i). \qquad (3.22)
\end{aligned}
$$

In the similar way, we will have:

$$
dif_{ep\_II}(n,i) = E\left\{\hat{f'}^{i}_{n} - \tilde{f'}^{i}_{n-1}\right\} = (\hat{f'}^{i}_{n} - \hat{f'}^{i}_{n-1}) + dif_{ep}(n-1,i). \qquad (3.23)
$$

CASE III: The co-located pixel in previous frame was Intra coded and was received at the decoder. So, the decoder acts the same as the encoder:

$$
d'_{ep\_III}(n,i) = E\left\{\left(\hat{f'}^{i}_{n} - \tilde{f}^{i}_{n-1}\right)^2\right\} = E\left\{\left(\hat{f}^{i}_{n-1} - \tilde{f}^{i}_{n-1}\right)^2\right\} = 0 \qquad (3.24)
$$

$$
dif_{ep\_III}(n,i) = E\left\{\hat{f'}^{i}_{n} - \tilde{f}^{i}_{n-1}\right\} = E\left\{\hat{f}^{i}_{n-1} - \tilde{f}^{i}_{n-1}\right\} = 0. \qquad (3.25)
$$

CASE IV: In this case, f^{i}_{n-1} was coded as Intra, but since it was lost during the transmission, the decoder considers that as an Inter coded pixel. We will have:

$$
\begin{aligned}
d'_{ep\_IV}(n,i) &= E\left\{\left(\hat{f'}^{i}_{n} - \left((\alpha-\beta)\tilde{f}^{i}_{n} + (1-\alpha)\tilde{f'}^{i}_{n-1} + \beta\tilde{K}^{i}_{n}\right)\right)^2\right\} \\
&= E\left\{\left(\hat{f}^{i}_{n-1} - \left((\alpha-\beta)\tilde{f}^{i}_{n} + (1-\alpha)\tilde{f'}^{i}_{n-1} + \beta\tilde{K}^{i}_{n}\right)\right.\right. \\
&\qquad \left.\left. +(\alpha-\beta)\left(\hat{f}^{i}_{n} - \hat{f}^{i}_{n}\right) + (1-\alpha)(\hat{f'}^{i}_{n-1} - \hat{f'}^{i}_{n-1})\right)^2\right\} \\
&= E\left\{\left(\hat{f}^{i}_{n-1} - \left((\alpha-\beta)\hat{f}^{i}_{n} + (1-\alpha)\hat{f'}^{i}_{n-1} + \beta\hat{K}^{i}_{n}\right)\right.\right. \\
&\qquad \left.\left. +(\alpha-\beta)\left(\hat{f}^{i}_{n} - \tilde{f}^{i}_{n}\right) + (1-\alpha)(\hat{f'}^{i}_{n-1} - \tilde{f'}^{i}_{n-1})\right)^2\right\} \qquad (3.26)
\end{aligned}
$$

$$
\begin{aligned}
&= E\Big\{\Big(\hat{f}^i_{n-1} - \big((\alpha-\beta)\hat{f}^i_n + (1-\alpha)\hat{f'}^i_{n-1} + \beta\hat{K}^i_n\big) \\
&\qquad +(\alpha-\beta)\big(\hat{f'}^j_{ref} + \hat{r}^i_n - \tilde{f'}^j_{ref} - \hat{r}^i_n\big) + (1-\alpha)(\hat{f'}^i_{n-1} - \tilde{f'}^i_{n-1})\Big)^2\Big\} \\
&= E\Big\{\Big(\hat{f}^i_{n-1} - \big((\alpha-\beta)\hat{f}^i_n + (1-\alpha)\hat{f'}^i_{n-1} + \beta\hat{K}^i_n\big) \\
&\qquad +(\alpha-\beta)\big(\hat{f'}^j_{ref} - \tilde{f'}^j_{ref}\big) + (1-\alpha)(\hat{f'}^i_{n-1} - \tilde{f'}^i_{n-1})\Big)^2\Big\} \\
&= \Big(\hat{f}^i_{n-1} - \big((\alpha-\beta)\hat{f}^i_n + (1-\alpha)\hat{f'}^i_{n-1} + \beta\hat{K}^i_n\big)\Big)^2 \\
&\quad +(\alpha-\beta)^2 d'_{ep}(ref,j) + (1-\alpha)^2 d'_{ep}(n-1,i) \\
&\quad +2(\alpha-\beta)(1-\alpha)dif_{ep}(n-1,i)dif_{ep}(ref,j) \\
&\quad +2\Big(\hat{f}^i_{n-1} - \big((\alpha-\beta)\hat{f}^i_n + (1-\alpha)\hat{f'}^i_{n-1} + \beta\hat{K}^i_n\big)\Big) \\
&\qquad \times\Big((1-\alpha)dif_{ep}(n-1,i) + (\alpha-\beta)dif_{ep}(ref,j)\Big).
\end{aligned} \tag{3.27}
$$

Eq. (3.26) is based on assumption that the difference between the average $\hat{K}^i_n$ and $\tilde{K}^i_n$ can be neglected (Table 3.1). Using similar derivations, we obtain:

$$
\begin{aligned}
dif_{ep\_IV}(n,i) &= E\Big\{\hat{f'}^i_n - \big((\alpha-\beta)\tilde{f}^i_n + (1-\alpha)\tilde{f'}^i_{n-1} + \beta\tilde{K}^i_n\big)\Big\} \\
&= \Big(\hat{f}^i_{n-1} - \big((\alpha-\beta)\hat{f}^i_n + (1-\alpha)\hat{f'}^i_{n-1} + \beta\hat{K}^i_n\big)\Big) \\
&\quad +(1-\alpha)dif_{ep}(n-1,i) + (\alpha-\beta)dif_{ep}(ref,j).
\end{aligned} \tag{3.28}
$$

By considering the four different cases (Eq. (3.20)-Eq. (3.28)), $d'_{ep}(n,i)$ and $dif_{ep}(n,i)$ are calculated respectively as:

$$
\begin{aligned}
d'_{ep}(n,i) &= E\Big\{\Big(\hat{f'}^i_n - \tilde{f'}^i_n\Big)^2\Big\} \\
&= \begin{cases}
p^2 d'_{ep\_II}(n,i) + p(1-p)d'_{ep\_IV}(n,i) & \text{if } f^i_{n-1} \text{ is Intra,} \\
(1-p)d'_{ep\_I}(n,i) + pd'_{ep\_II}(n,i) & \text{otherwise.}
\end{cases}
\end{aligned} \tag{3.29}
$$

$$
\begin{aligned}
dif_{ep}(n,i) &= E\left\{ \hat{f'}_n^i - \tilde{f'}_n^i \right\} \\
&= \begin{cases} p^2 dif_{ep\_II}(n,i) + p(1-p) dif_{ep\_IV}(n,i) & \text{if } f_{n-1}^i \text{ is Intra,} \\ (1-p) dif_{ep\_I}(n,i) + p dif_{ep\_II}(n,i) & \text{otherwise.} \end{cases}
\end{aligned} \tag{3.30}
$$

It should be noted that, when a block is coded as Intra, $d'_{ep}(n,i)$ is equal to zero. In case of bi-prediction Inter coding, we will have:

$$
d'_{ep}(ref,j) = w_0 d'_{ep}(n\_ref0, j0) + w_1 d'_{ep}(n\_ref1, j1), \tag{3.31}
$$

where w_0 and w_1 are the weights of reference 0 (n_ref0) and reference 1 (n_ref1) used in bidirectional video coding. $j0$ and $j1$ denote to reference pixels in reference 0 and reference 1.

3.4 Simulation Results

To study the performance of the proposed technique, a set of simulations were conducted with commonly used conditions for error resilience simulations [120]. The employed conditions are:

- JSVM 9.15 [121] was used as the SVC encoder and decoder.
- Standard sequences, "Akiyo", "Bus", "Flower", "Football", "Foreman", "News", "Mobile", "Paris" and "Stefan" were encoded with QCIF and CIF @ 30fps and "City", "Crew", "Ice", "Harbour" were coded with CIF and 4CIF @ 30fps in the simulations.
- There were four temporal, and two spatial layers in each coded stream. 25% of the total bit rate is allocated for the base spatial layer, and the rest is used for encoding the second spatial layer. In each spatial layer, the initial QPs of the lowest to the highest temporal layers changes by -2, 1, 3 and 4 respectively.
- Quality scalability was not employed.

- To get a fixed bit rate, the "FixedQPEncoderStatic" tool in JSVM was used. This tool finds the appropriate QP for the target bit rate.

- Multiple slicing was used. In order to have similar slice loss patterns in all the techniques, frames are divided into same number of slices. So, each slice contains one row of MBs in QCIF, 2 rows of MBs in CIF and 4 rows of MBs in 4CIF sizes. Each packet includes one slice, which forms an RTP packet. Based on the assumption that RTP/UDP/IP transmission is used, lost or damaged packets are discarded without retransmission.

- In order to simulate a network with losses, four packet loss patterns included in ITU-T VCEG [122, 123] were employed. The average packet loss rates (PLR) are 1%, 3%, 5%, and 10%. These patterns were obtained by experiments on the Internet backbone [123].

- Encoded sequences were repeated 40 times and transmitted through a packet loss channel to get more consistent results.

- In order to observe the performance of different RFM particularly, we used two error concealment techniques at the decoder side. First, a simple yet effective Picture Copy (PC) was used. In this technique, each loss slice is concealed by copying the co-located slice from the reference frame 0. Also, we used Motion Copy or Motion Compensated (MC) temporal concealment as a more complex error concealment technique. In this technique, missing areas are reconstructed by motion compensation using the reference frame 0 [63].

- The transmitted video was decoded and the average peak signal to noise ratio (PSNR) was calculated over all pictures.

Usually, RFM methods are combined with Intra refreshing techniques to get better performance. In order to observe the performance of different reference frame modification methods particularly, the simplest Intra updating method, which is random Intra refresh, was used in the first part of our simulations. In the second part, the end-to-end distortion estimation is used in mode decision of our proposed tehcnique.

3.4.1 Performance of the proposed prediction structures

Rate distortion curves of different methods for "Foreman", "Mobile", "City', and "Crew" sequences are shown in Fig. 3.10 and Fig. 3.11. For "Foreman" and "Mobile" sequences, the base and enhancement spatial layers are encoded with QCIF and CIF sizes, and for "City" and "Crew" sequences, the base and enhancement spatial layers are encoded with CIF and 4CIF sizes. For each sequence, results of picture copy and motion copy as error concealment technique are shown in Fig. 3.10 and Fig. 3.11, respectively. Encoded videos are transmitted over a channel with 5% packet loss. α is calculated based on Eq. (2.28) and and assuming $H = 0.1$ for all the techniques. w_0 in the first proposed prediction structure is set to 0.7 and we set β in the second proposed method to 0.2.

It can be observed that increasing the bit rate will result in better quality at the decoder. Also, using RFM techniques achieves better performance compared to the normal video coding. As it was reported in [33, 34], IGSCP performs better than the GSCP and Normal methods. It admits the effect of exploiting Intra MBs in increasing the robustness of the video stream. In addition, the first proposed prediction structure will result in a higher PSNR compared to the GSCP and IGSCP techniques in most of the cases. It can be noticed that the gain of the proposed method over IGSCP is more visible as bit rate increases. The reason is that the proposed technique exploits the Intra MBs more efficiently than IGSCP, and in higher bit rates more macroblocks are encoded as Intra, therefore the improvement is more significant. The gain of our first technique over IGSCP varies for different sequences. We get a higher gain in sequences like "Crew", "Foreman", "Bus" and "Paris", while the gain is not significant in sequences like "Mobile", "City" [74]. The main reason that we observe different behaviours of this technique is because of using a fixed value for w_0. By tuning this coefficient for different sequences and bit rates, the quality of the decoded would improve, but this might not be practical in some applications.

The second proposed technique outperforms other three reference modification techniques in all bit rates. As it can be noticed, the gain of this technique over other methods would be more noticeable by increasing the bit rate. The reason is that at high bit rates, the number of Inter MBs is smaller, and the negative effect of using leaky prediction on predictive coding is less considerable.

Furthermore, as we expected, using motion compensated (MC) instead of picture copy (PC) as the error concealment technique would give us a better performance for normal

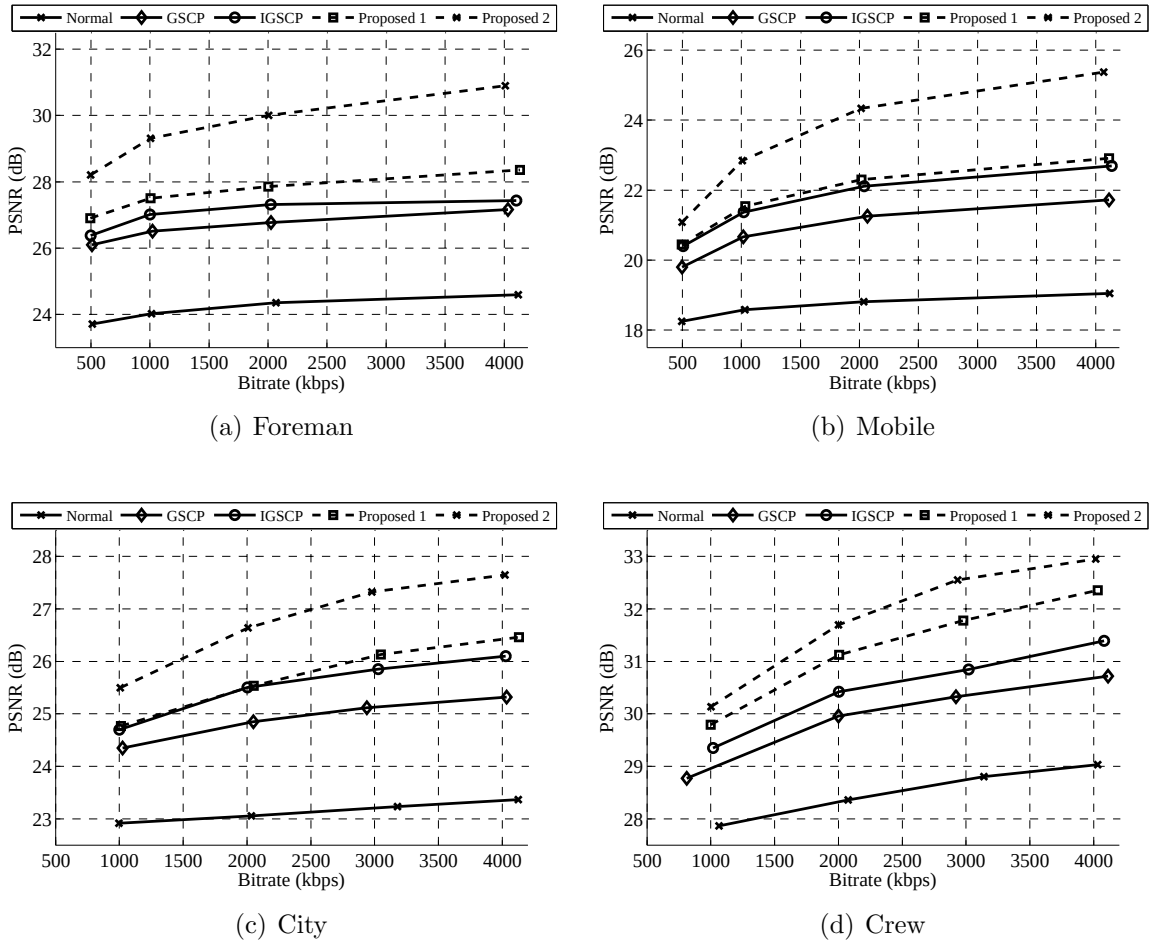


Fig. 3.10 Rate distortion curves for different methods for (a) "Foreman" sequence, (b) "Mobile" sequence, (c) "City" sequence, (d) "Crew" sequence. Packet loss rates of 5% for both base and enhancement layers. GoP size of 8. Picture copy used as the error concealment technique.

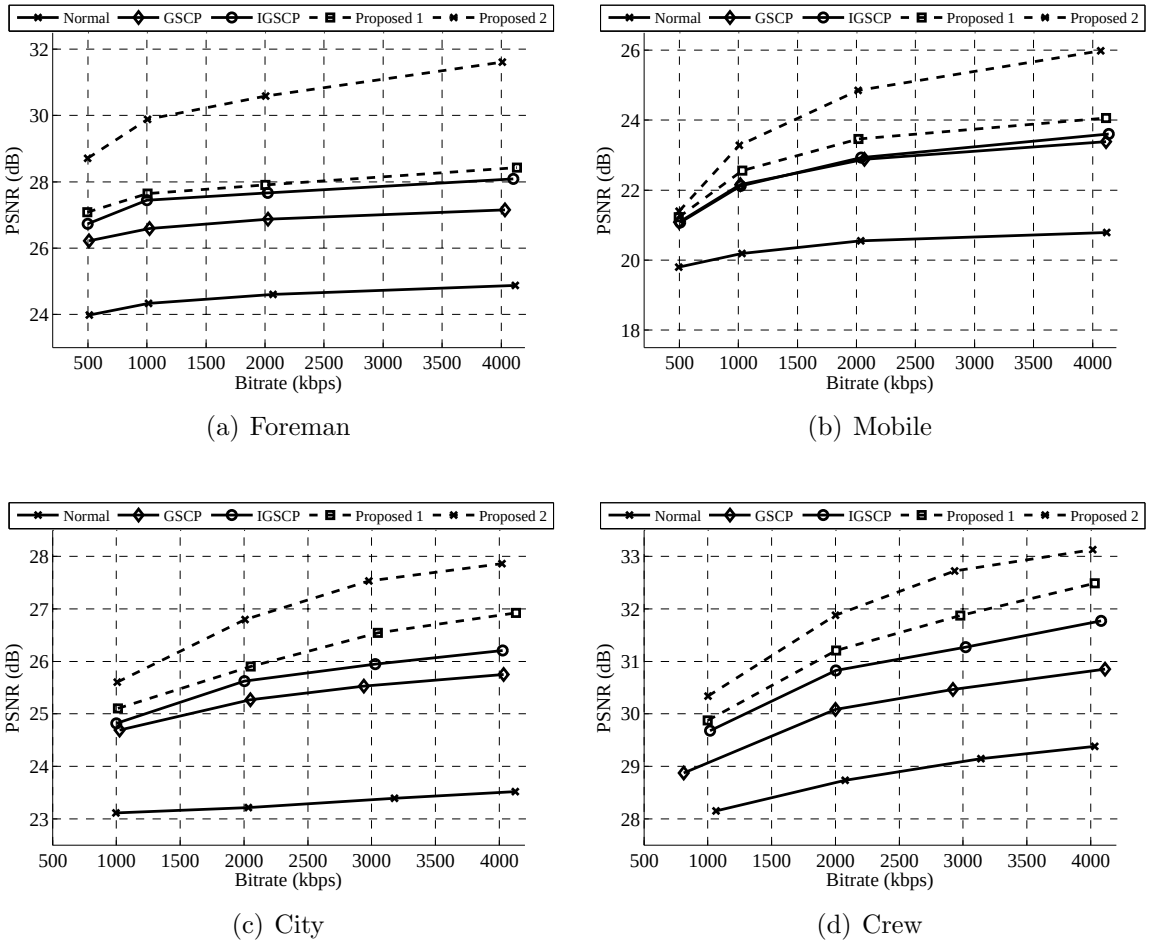


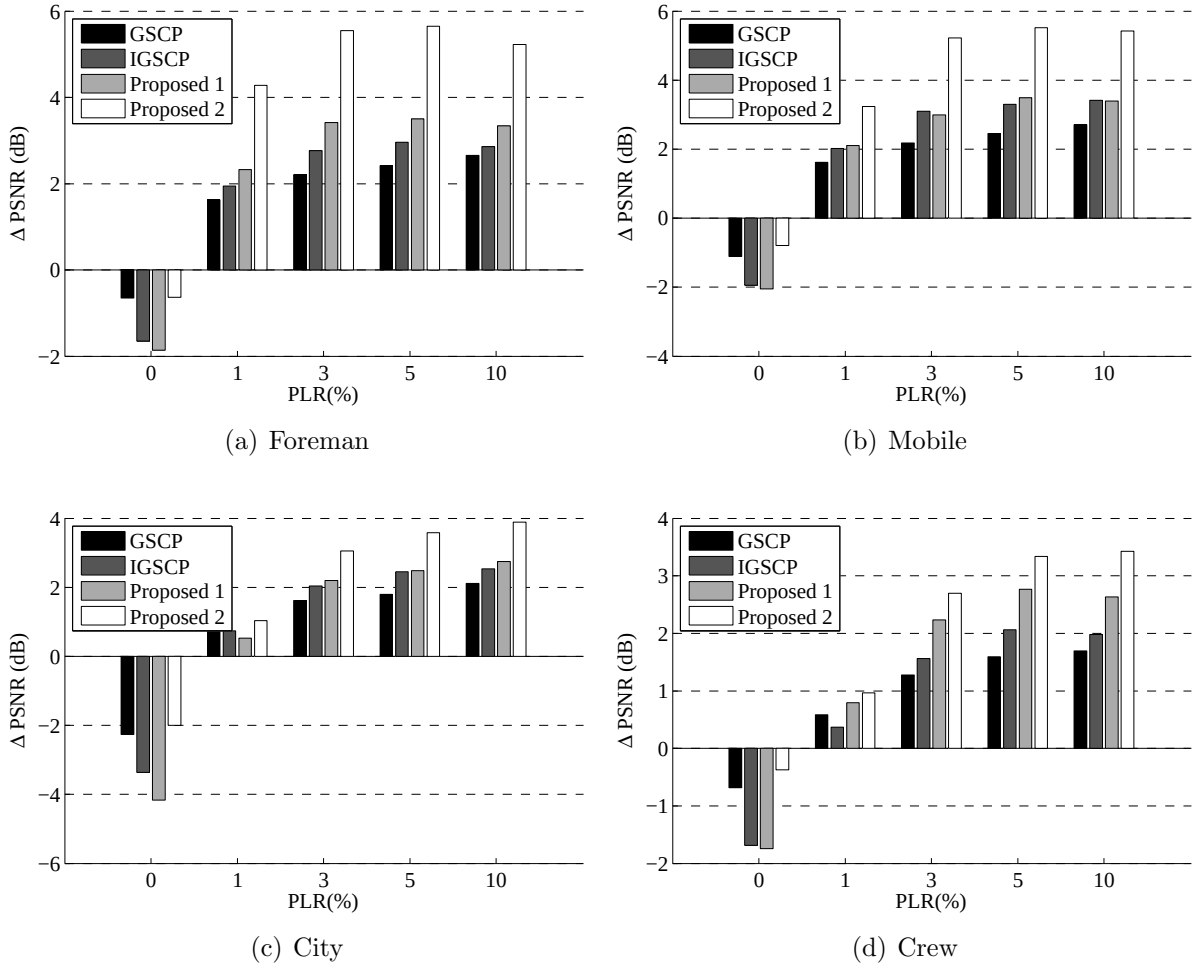
Fig. 3.11 Rate distortion curves for different methods for (a) "Foreman" sequence, (b) "Mobile" sequence, (c) "City" sequence, (d) "Crew" sequence. Packet loss rates of 5% for both base and enhancement layers. GoP size of 8. Motion copy used as the error concealment technique.

coding (gains of up to 2 dB). But, since we are using a GoP size of 8, the gain of using motion compensated concealment over simple picture copy is much less than the gain in the IPPP structure. It is because of the temporal distance between the current frame and the reference frame, which is only 1 for the last temporal layers but can be up to 8 frames in the other layers [63]. Also, since only the first frame of the whole sequence is encoded as Intra frame, the error would propagate to future frames unless a block is coded as Intra. This is another reason that using MC is not performing much better compared to PC.

We have assumed that all the RFM techniques do not have any knowledge of the error concealment technique used at the decoder side, and the same structures explained in Section 2.6, Section 3.2.1 and Section 3.2.2 are used. Although we expected gains by replacing the error concealment technique in RFM methods, we observed that using MC instead of PC as the error concealment method sometimes leads to slight losses in the RFM techniques and the proposed methods. As a result, the gain of the second proposed method compared to the normal coding can drop up by to 2 dB by using motion compensated concealment technique.

Fig. 3.12 and Fig. 3.13 illustrate the PSNR differences between each method and the normal coding at five different packet loss rates for using picture copy and motion copy respectively. It can be observed naturally that utilizing reference frame modification techniques reduces the coding performance in error free case. Our first method does not consistently outperform previous methods. In some cases, IGSCP achieves gains (up to 0.6 dB) over our first method. In these cases, exploiting the Intra MBs of both references leads to reduction in coding efficiency. However, using the spatial local average in the second proposed technique mitigates this negative impact. As a result, our second proposed technique, performs better than other reference frame modification techniques, but still worse than the normal prediction structure in error free cases.

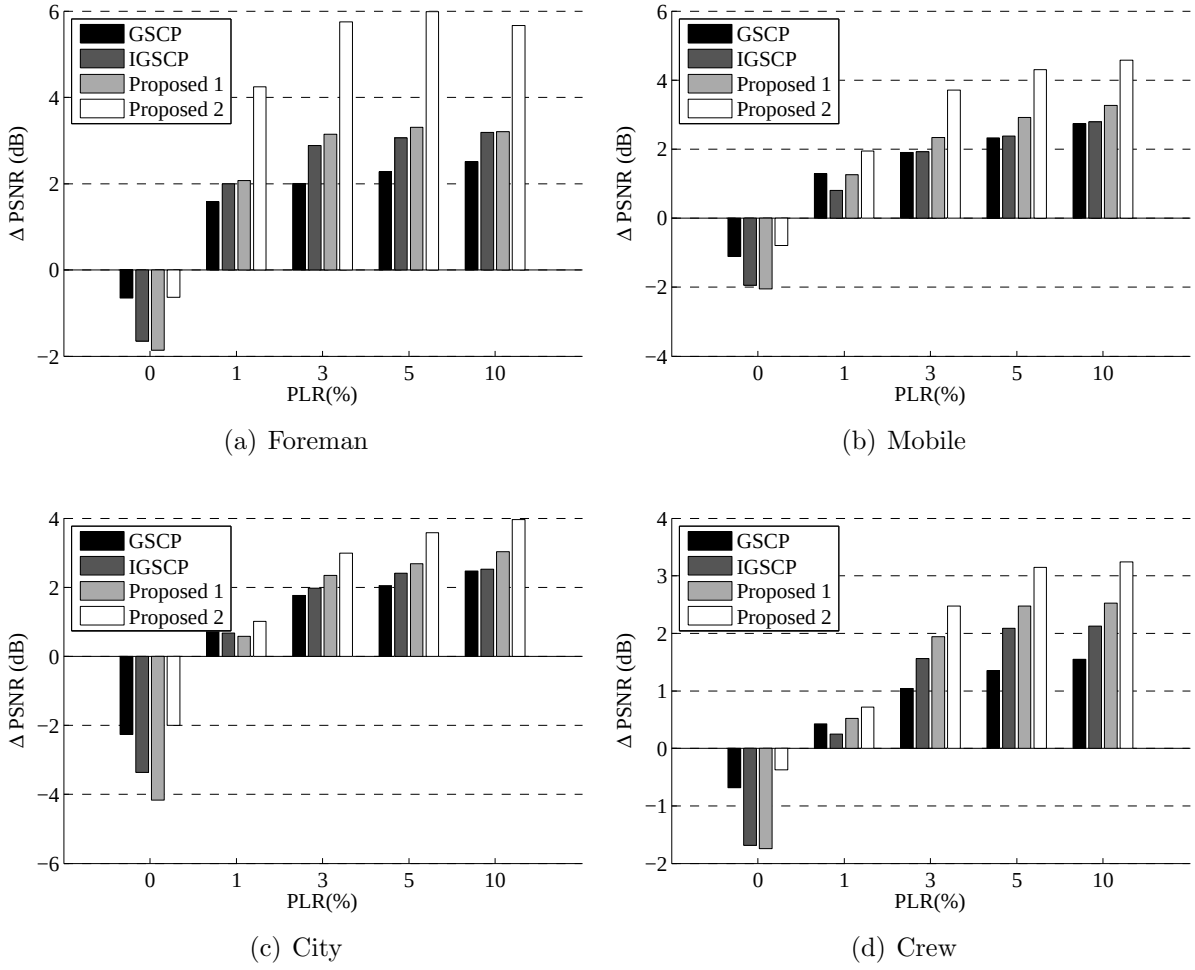
In error prone channels, using RFM techniques always results in more resilient streams compared to the normal coding. These sets of figures confirm that our second proposed method performs better than previous ones at different bit rates and channel conditions. Using picture copy as the error concealment technique, on average, over all tested sequences, different bit rates and various packet loss rates, we observed average gains of 1.27 dB, 1.83 DB and 3.65 dB over the first proposed technique, IGSCP and normal coding respectively. Using the MC technique as the error concealment, the gains were 0.96 dB, 1.55 dB and 3.36 dB respectively. It should be mentioned that although using the spatial local average



(a) Foreman (b) Mobile

(c) City (d) Crew

Fig. 3.12 Δ PSNR of different RFM methods over normal coding at various packet loss rates for (a) "Foreman" sequence, (b) "Mobile" sequence, (c) "City" sequence, (d) "Crew" sequence. Encoded at 2048 kbps and GoP size of 8. Picture copy used as the error concealment technique.



(a) Foreman (b) Mobile

(c) City (d) Crew

Fig. 3.13 Δ PSNR of different RFM methods over normal coding at various packet loss rates for (a) "Foreman" sequence, (b) "Mobile" sequence, (c) "City" sequence, (d) "Crew" sequence. Encoded at 2048 kbps and GoP size of 8. Motion copy used as the error concealment technique.

reduces the negative impact of leaky prediction on predictive coding significantly, it was observed that the achieved gains are smaller in lower bit rates and slow sequences such as “News” and “Akiyo”. As we discussed before, this is because of the noticeable role of temporal prediction in these cases.

3.4.2 Reference frame modification with error resilience mode decision

Rate distortion curves of different methods for “Foreman”, “Mobile”, “City” and “Crew” sequences are depicted in Fig. 3.14. For “Foreman” and “Mobile” sequences, the base and enhancement spatial layers are encoded with QCIF and CIF sizes, and for “City” and “Crew” sequences, the base and enhancement spatial layers are encoded with CIF and 4CIF sizes. The coded streams are transmitted over a channel with 5% packet loss. The performance of the second proposed method is shown in combination with random Intra refresh (RIR) with 10% Intra rate and using the estimated end-to-end distortion in the mode decision (LARDO techniques).

As it was illustrated in the previous section, the proposed method performs better than the normal coding. Also using the LARDO technique achieves better performance compared to the random Intra refresh method. This is because of using the estimated end-to-end distortion in the selection of the best mode. Furthermore, by increasing the bit rates, the two methods using LARDO perform better than other techniques. The reason is that LARDO adds more optimally selected Intra MBs in higher bit rates. The PSNR differences between these three techniques and the normal coding at five different packet loss rates are illustrated in Fig. 3.16. The sequences are encoded at 30 fps and bit rate of 2048 kbps. It can be observed that utilizing the LARDO technique does not reduce the coding performance in error free cases. By increasing the packet loss rate, the performance of using the error resilience mode decision over random Intra refresh is more significant. Furthermore, making use of the proposed method in combination with LARDO, improves the performance of the LARDO technique. On average, a gain of up to 0.80 dB over LARDO can be achieved. In addition, using LARDO in combination with the proposed technique instead of random Intra refresh would give an average gain of 2.72 dB.

It should be mentioned that LARDO achieves a good error resilience performance by optimally increasing the number of Intra MBs. These Intra MBs do not use temporal prediction from previous frames. As a result, the reference frame modification structure is

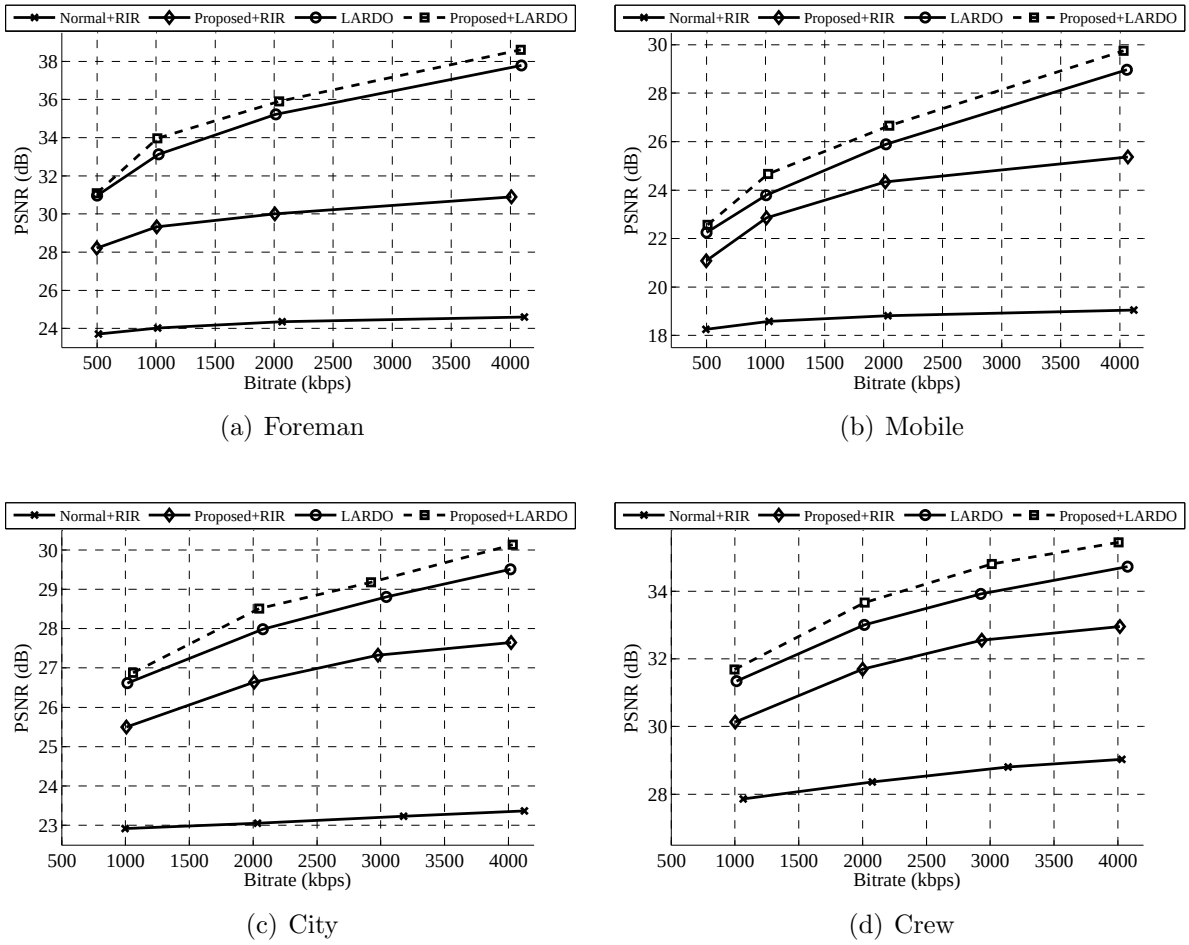
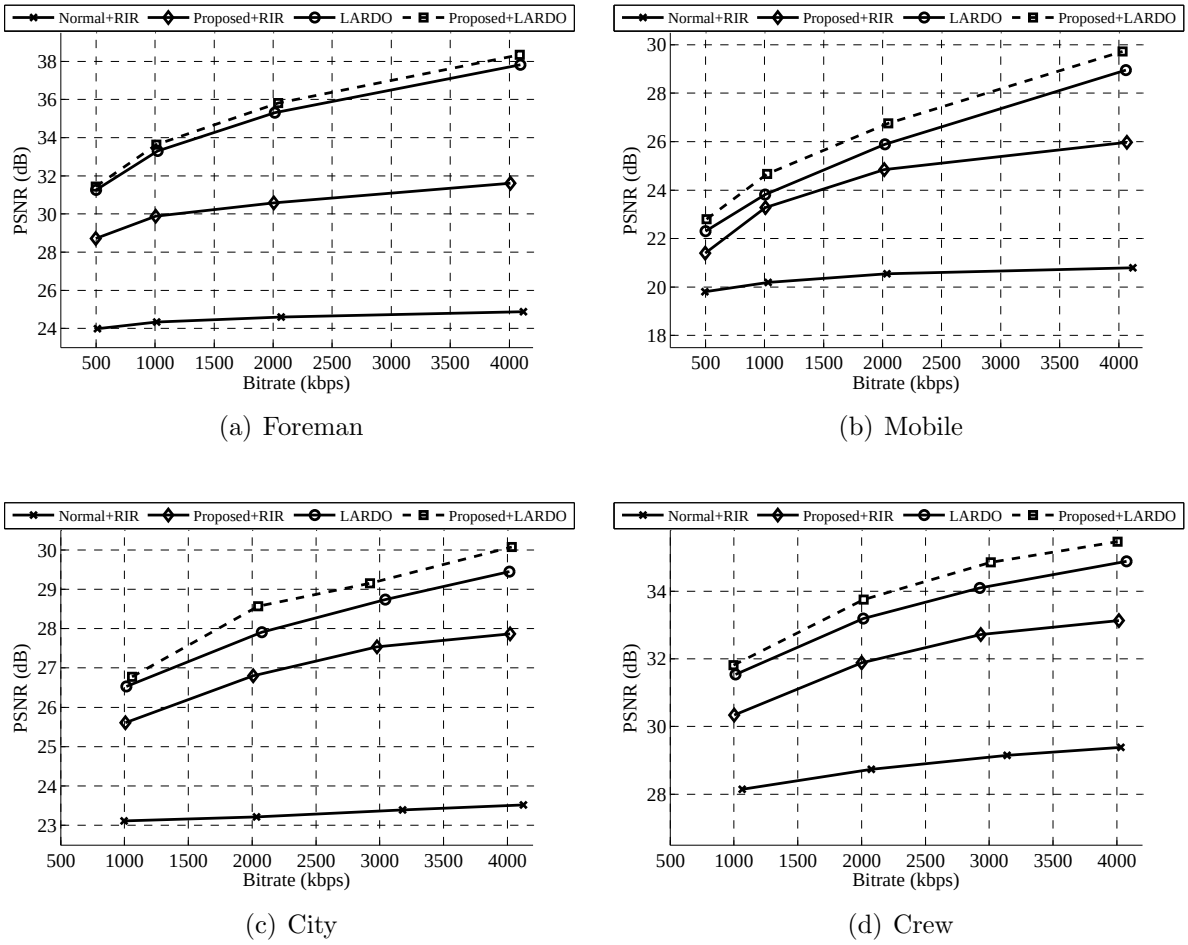


Fig. 3.14 Rate distortion curves for different methods for (a) "Foreman" sequence, (b) "Mobile" sequence, (c) "City" sequence, (d) "Crew" sequence. Packet loss rates of 5% for both base and enhancement layers. GoP size of 8. Picture copy used as the error concealment technique.



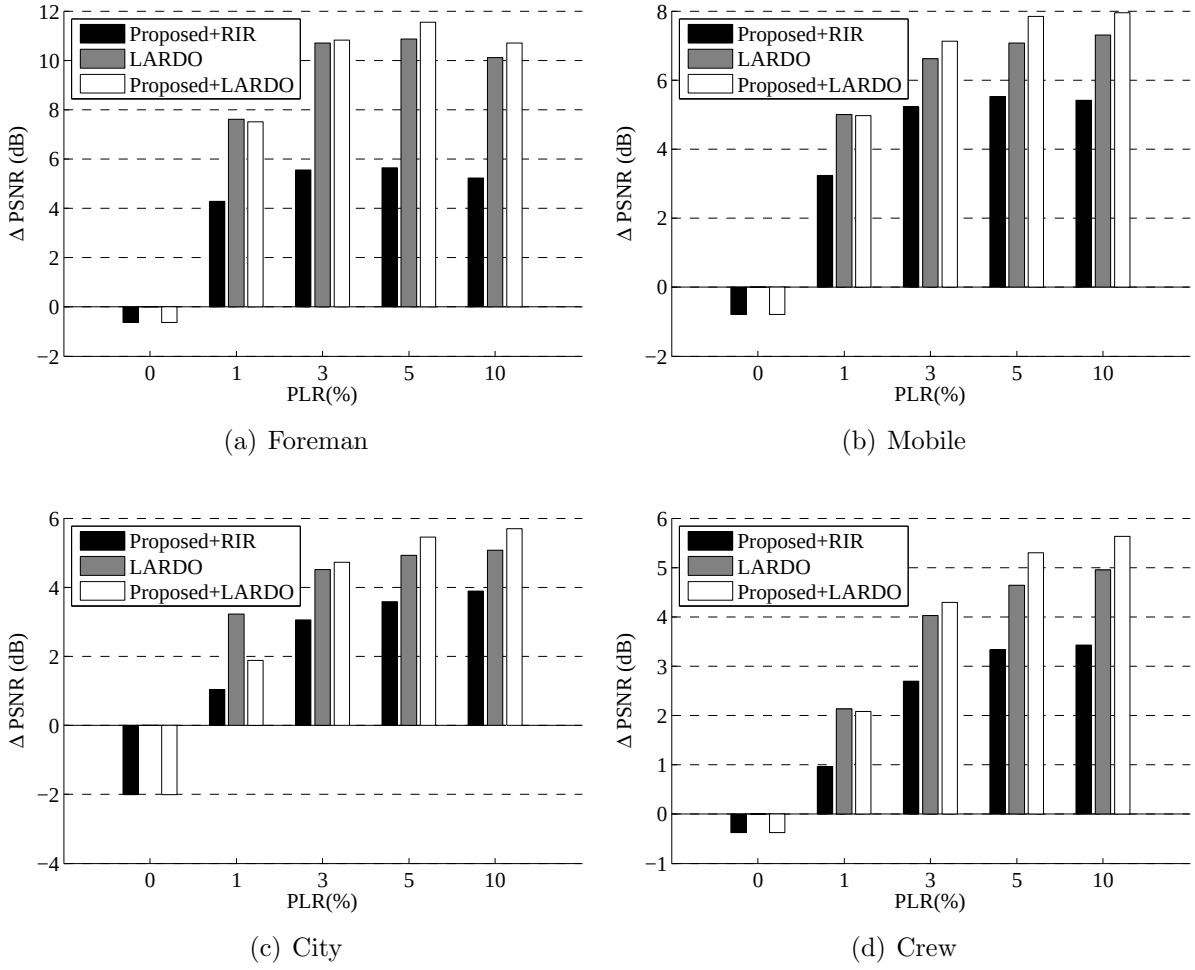
(a) Foreman

(b) Mobile

(c) City

(d) Crew

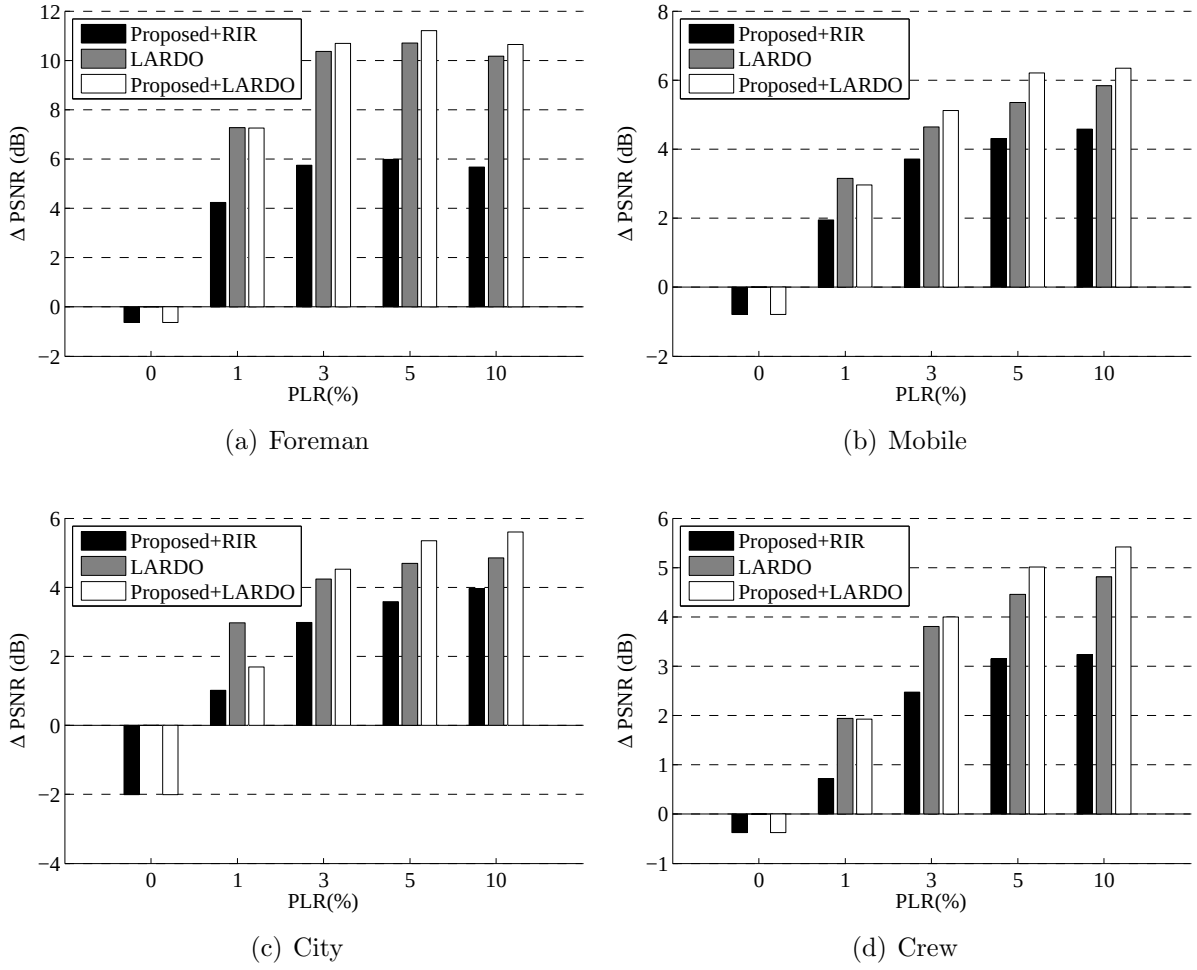
Fig. 3.15 Rate distortion curves for different methods for (a) "Foreman" sequence, (b) "Mobile" sequence, (c) "City" sequence, (d) "Crew" sequence. Packet loss rates of 5% for both base and enhancement layers. GoP size of 8. Motion copy used as the error concealment technique.



(a) Foreman (b) Mobile

(c) City (d) Crew

Fig. 3.16 Δ PSNR of different RFM methods over normal coding at various packet loss rates for (a) "Foreman" sequence, (b) "Mobile" sequence, (c) "City" sequence, (d) "Crew" sequence. Encoded at 2048 kbps and GoP size of 8. Picture copy used as the error concealment technique.



(a) Foreman (b) Mobile

(c) City (d) Crew

Fig. 3.17 Δ PSNR of different RFM methods over normal coding at various packet loss rates for (a) "Foreman" sequence, (b) "Mobile" sequence, (c) "City" sequence, (d) "Crew" sequence. Encoded at 2048 kbps and GoP size of 8. Motion copy used as the error concealment technique.

not effective on these blocks. We observed that an increase in the number of Intra MBs, decreases the improvement of our proposed method. For instance, the gain of our technique over LARDO is less in sequences with more Intra blocks ("Stefan") compared to sequences with fewer Intra blocks ("Mobile"). Furthermore, by increasing the GoP size, the temporal distance between the current frame and the reference frame raises. Consequently, the error concealment ($d_{ec}(n,i)$) and error propagation ($d_{ep}(n,i)$) distortions from previous frame used in Eq. (2.17) will increase. This will result in selecting more blocks as Intra, which do not propagate error from previous frames, by the LARDO mode decision process. Table 3.3 shows the average percentage of Intra coded MBs for different sequences by changing the GoP size. These video streams were encoded with the LARDO technique at QP of 28 and by assuming a packet loss rate of 10%. It can be noticed that more MBs are Intra coded in higher GoP sizes. This will justify our observation that by increasing the GoP size, the gain of using the proposed technique over LARDO decreases. In higher GoP sizes, LARDO encodes fewer blocks as Inter. As a result, our proposed technique is employed in a smaller number of MBs.

It should be mentioned that all these error resilience mode decision techniques require a input from the channel to estimate the packet loss rate. This information might not be available in many applications where these techniques are not practical. Furthermore, since p is used in many of the above equations, using an inaccurate packet loss rate will change the estimated end-to-end distortion significantly. Consequently, the mode selection process is affected and a non-optimal mode might be selected. This will result in lower performance.

In order to provide a better representation, a subjective comparison of different techniques is illustrated in Fig. 3.18. The images represent the decoded frame number 51 of Foreman sequence. The sequences were encoded at 1024 kbps with CIF size and 30 fps. Encoded videos were transmitted over a channel with 10% packet loss and same loss pattern was used for all the five techniques. Five temporal and two spatial layers are included in each stream. The proposed technique improves the quality of the received video by making use of the introduced leaky structure and making use of Intra MBs efficiently.

(a) Normal

(b) GSCP

(c) IGSCP

(d) The first proposed method

(e) The second proposed method

Fig. 3.18 Subjective results for "Foreman" sequence frame number 51 with CIF size and 30 fps at 1024 kbps, packet loss rate of 10%.

Table 3.3 Average percentage of Intra coded MBs in each frame for different sequences coded by the LARDO technique at 10% packet loss rate and QP =28.

Sequence	GoP=1	GoP=2	GoP=4	GoP=8	GoP=16
Akiyo	2.07	4.28	5.03	6.68	7.09
Bus	66.84	82.90	80.51	83.53	85.59
Flower	55.00	65.18	63.16	64.76	65.23
Football	66.76	78.75	78.62	81.59	83.37
Foreman	25.34	45.87	51.90	59.89	66.23
Mobile	25.81	61.67	69.40	78.14	81.09
News	6.07	10.81	12.15	14.71	16.18
Paris	81.42	94.64	94.46	94.36	93.84
Stefan	46.10	60.53	61.81	66.50	71.70

3.4.3 Computational Complexity

All the reference frame modification techniques need an extra buffer compared to the normal video coding in order to store the modified reference frame. In terms of the computational complexity, all these techniques only add a few additions and multiplications that are negligible compared to encoder components like motion estimation and mode decision. Furthermore, the second proposed method requires a process to calculate the local spatial average which can be implemented in an optimized way. In addition, as it was mentioned before, the modified reconstructed frame values, similar to reconstructed frame, are represented in integer values. Consequently, there is no extra complexity in motion estimation and motion compensation.

In order to get an estimate of the added computation, we measured the encoding and decoding times for each method. On average over all sequences and bit rates, the encoding and decoding times increased about 6.4% and 11.9%, respectively, compared to the normal coding. It should be mentioned that the additional processing time achieves an average gain of 3.65 dB. The added encoding and decoding times of the proposed method compared to the IGSCP technique are about 2.9% and 6.6% respectively. All the simulations are conducted on a PC with Intel Core i7-2600 Processor (8M Cache, 3.40 GHz) and 8 GB of RAM, and no specific speed optimization was carried out on the code.

3.5 Chapter Summary

In order to mitigate the impact of transmission errors on the quality of delivered video, different error resilience techniques have been proposed. In this chapter, we extended the previous reference frame modification methods to temporal and spatial scalability of the scalable extension of H.264/AVC. We presented two new reference frame modification techniques. The first proposed technique improves the previous methods by exploiting the Intra MBs in reference frames efficiently.

The second proposed technique exploited a new leaky prediction structure in addition to efficiently making use of the Intra MBs in reference frames. It jointly makes use of (i) error robustness of previous Intra MBs, (ii) good prediction resulting from using the previous reference frame, and (iii) exponential decay of error propagation caused by leaky prediction. It was observed that the video quality was increased especially for medium and high motion sequences. On average, gains of 1.27 dB and 0.96 dB over the previous method with picture copy, and motion compensated error concealment methods respectively. Employing this method increased the average processing time of a normal encoder by 12%, while the average quality improvement over normal coding was about 3.5 dB, which is a significant gain.

In order to further improve the error robustness, the end-to-end distortion of the second proposed prediction structure was calculated and used in the mode decision process. In this way, the best mode is selected based on compression efficiency and error resilience. By using the estimated end-to-end distortion in the mode decision process, we improved the performance of our technique by 2.72 dB on average. It should be mentioned that the end-to-end distortion estimation requires an estimate of the channel packet loss rate which might not be available in some applications.

In the next chapter, we are going to focus on the utility estimation of different layers in an scalable stream. The estimated utilities can be used to quantify the importance of each layer, which is required in unequal error protection techniques.

Chapter 4

Utility Calculation for Unequal Error Protection

Another approach to address the problem of video transmission over error prone networks is protecting the coded sequence with forward error correction (FEC) codes. Since each video packet has different contribution to the video quality and different sensitivity to packet losses, unequal error protection (UEP) can be applied on video signals. The idea of UEP is to protect each video part differently based on its importance. Applying UEP on different types of scalable coded video has been explored by many researchers [35–42, 124–127]. One of the key components in these techniques is to analyse the utility of each video part. The utility of a frame or a layer is a metric that quantifies the quality contribution and importance of that frame or layer. A more accurate metric would result in a more effective protection, and consequently, would improve the quality of the transmitted video significantly.

Several techniques have been proposed in order to calculate the utility of each frame or slice. Most of these techniques are based on multiple decoding of sub bitstreams at the encoder side. In these methods, each video layer or frame is discarded from the main bitstream, decoded and the distortion or PSNR is calculated. Based on the calculated distortion, the total distortion and the distortion of previous parts, the utility is calculated. This process can be done at a frame level [35, 36], a layer level [38, 125–127] or a network abstraction layer (NAL) unit [40] level. The lower level this process is performed at, the more computationally complex the process is and the more accurately the utility is

calculated. An accurate utility calculation usually requires huge computation which is not practical in many applications like real time video streaming. To the best of our knowledge, a few papers have addressed this problem by proposing utility estimation techniques [39,42]. These techniques, which are proposed for quality scalability, are either very video content dependent or still computationally complex.

In this chapter, we propose a low complexity utility calculation technique. This technique does not require multiple decoding at the encoder and can be easily incorporated to the encoder. It estimates the utility of each block within a NAL unit by using the source coding distortion, error concealment distortion and error propagation to future frames. Despite the fact that end-to-end distortion calculation schemes [25, 26, 28, 29] are highly dependent on packet loss probability of the channel, the proposed method does not require the packet loss probability in order to calculate the utility. This method is extended to different prediction structure and temporal and spatial scalable video. Also, we propose a low delay version of the estimation technique which achieves a slightly lower performance, but can be utilized in applications with delay constraints. In order to evaluate our technique, we use a typical framework. Our utility estimation technique can be used in other frameworks with different protection codes, packetization and optimization techniques

We begin by presenting the existing utility estimation techniques. We then proceed by introducing and explaining our framework and the problem formulation. Then, the proposed utility calculation technique is introduced. Finally, we provide simulation results for all our proposed schemes showing the performance.

4.1 Unequal Error Protection Techniques

One of the approaches to protect the coded video against transmission errors is to use forward error correction (FEC) codes. Since each video packet has different contribution to the video quality and different sensitivity to packet losses, unequal error protection (UEP) can be applied on video signals. The idea of UEP is to allocate the constraint channel bits to different video packets based on their importance. Applying UEP on single layer video coding has been addressed by many researchers. Unequal error protection based on importance of I, P and B frames for H.263 video has been proposed in [35]. [36] models the channel distortion for I, P and non-referenced B frames in order to evaluate the importance of each frame. In [37], UEP is combined with data partitioning tool in H.264/AVC and

each data partition is protected with different priorities. Furthermore, since the video layers in an SVC stream have different importance and quality improvements, applying UEP on a scalable video signal further improves the efficiency and reliability of the video transmission. The combination of UEP and SVC has been widely studied with various types of scalability [38–42, 125–128].

One of the main issues in all these techniques is to find a proper metric to evaluate the importance of each layer or frame. This metric is usually referred to as the utility of the layer. The utility of a layer should indicate the contribution of the layer in the overall quality of the video sequence and, also its sensitivity to transmission errors. more accurate metric would result in more effective protection, and consequently, would improve the quality of the transmitted video significantly. Based on the utility evaluation, we categorize the previous techniques into three groups, each being described in one of the following sections.

4.1.1 Utility calculation using multiple decoding per layer

In this group of references, the utility is calculated for each layer based on the difference of the PNSRs or distortions of two coded streams, one with and one without that layer. This calculation is done at the encoder side and is based on decoding of different sub streams. Utility calculation per layer means that for each layer, the sub stream is extracted, decoded and the distortion is calculated. For example, for a coded video with 16 frames, three spatial and four quality layers, 12 utilities are calculated. Based on different types of scalability, several techniques have adopted this utility calculation.

The impact of applying UEP on different layers of fine granular scalable (FGS) video has been studied in [38]. Two-dimensional unequal error protection for temporal and quality scalability is proposed in [125, 126]. In these techniques, the utility calculation is done at quality and temporal levels. [126] solves the rate allocation problem by using a genetic algorithm, and in [125], the channel rate allocation is based on another evolutionary algorithm called Particle Swarm Optimization (PSO) [129]. [127] extends the idea of applying UEP on scalable coded video to the medium granularity scalability (MGS) of the standard, by solving the channel rate allocation problem by introducing Lagrangian relaxation. The authors extended their work into multicasting in [41]. They achieved good overall performance for different user distributions. Their technique tries to assign the channel bit rate

in an optimal way such that a good trade-off on the video qualities to different users is achieved.

4.1.2 Utility calculation using multiple decoding per NAL unit

In the previous group, the utility is calculated at the layer level. It means that all the frames with the same temporal, quality or spatial indexes would get the same utility. This is based on the assumption that different frames within the same layer have similar distortion, which is not true for the real scalable coded video. In fact, the content of each frame, QP and the selected modes would change the distortion. In this group of techniques, the proposed technique calculates the utility per network abstraction layer (NAL) unit. Each NAL unit might belong to a frame, spatial, temporal and quality layers. For example, for the same coded stream as above, 192 utilities are calculated. In this way, a more accurate but more computationally complex utility metric is achieved. In [40], a FEC allocation optimization algorithm that considers the error drifting problem for temporal and inter layer prediction is proposed. This technique considers the quality improvement of each frame, but based on the application and the computational resources, it would modify the utility metric at layer level too. It should be noted that there would be a trade of between the accuracy of calculation and the computational complexity.

4.1.3 Utility estimation

The actual measurement of the utility of each layer or frame by using multiple decoding requires huge computation and is not practical in many applications, especially in live video streaming. In this group of techniques, an estimation of the utility is used in order to evaluate the importance of each layer. [39] proposes a simple performance metric called layer-weighted expected zone of error propagation (LW-EZEP) to quantify the effect of the utility of each layer. It allocates the channel bits for protection of different temporal and quality layers of the SVC extension of H.264/AVC jointly. The estimated utility is defined as:

$$\gamma_{(i,j)} = \frac{2^{T-C_1 \times i} - 1}{(1+j)^{C_2}}, \tag{4.1}$$

where j and i are indices of the temporal and quality layer, T is the maximum number

of temporal layers. C_1 and C_2 are temporal and quality layer scaling factors which are calculated empirically. The main drawback of this technique is the calculation of C_1 and C_2 which are calculated experimentally and might change for different video sequences, number of layers and bit rates.

Furthermore, in [42], the authors extended their previous work [127] and proposed a method to estimate the utilities of each enhancement layer in quality scalability. The estimation is used instead of multiple decoding, which avoids the added decoding time for all the enhancement layers. But in the base layer, multiple decoding is still used for utility calculation. It performs the multiple decoding process for one GoP and derives a set of parameters for the future GoPs. If this model is used in long sequences or sequences with scene changes and rapid movement, the estimation becomes significantly inaccurate.

4.2 Framework and Problem Formulation

After coding the video with a scalable encoder, the coded bitstream contains different temporal and spatial layers. Each of these layers has different importance and sensitivity to transmission errors. Furthermore, within each frame, there might be several slices. Each slice forms a video packet or NAL unit. Based on the video content and the size of the packet, it would contribute to the quality of the coded video. Two NAL units might be different in their slice index, frame number and spatial resolution. In order to protect this stream made up of several types of NAL units, unequal error protection can be used. In UEP, each NAL unit, which has a slice index, frame number and spatial layer index, is protected with different number of parity bits. In order to apply UEP and form transmission packets, the packetization scheme shown in Fig. 4.1 is used [130]. By using this scheme, each video packet or NAL unit can be protected independently.

Assuming the number of NAL units to be L, i, in the range of 1 and L, represents the index of each NAL unit. The i^{th} NAL unit with B_{S_i} source bits is protected with B_{C_i} parity bits. Furthermore, each row in Fig. 4.1 corresponds to a transmission packet. The total number of packets and the packet size are represented by N and M respectively. B_{S_i} and B_{C_i} are distributed into S_i and C_i packets with l_i symbols in each packet. In order to generate parity bits, Reed Solomon (RS) codes are usually employed. RS codes are widely used in applications like storage devices (DVD), mobile communications, and high speed modems. Reed Solomon codes, which are linear non binary block codes, result in

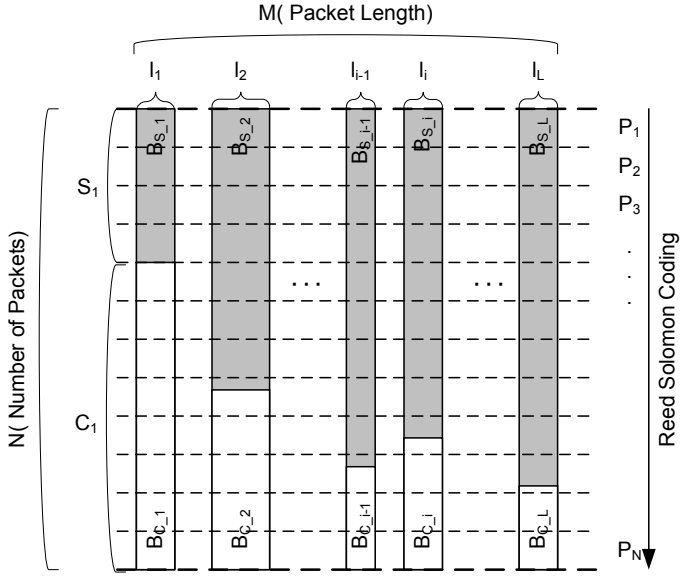


Fig. 4.1 Packetization scheme.

maximum erasure protection while adding the minimum redundancy [131]. Another group of erasure codes that can be used for network applications are Tornado codes. Tornado codes can be coded and decoded more efficiently compared to RS codes, but have slightly lower protecting properties [132]. Also, low density parity check (LDPC) codes [133] are other types of error correction codes that can be used in this framework. Since our focus in this chapter is only on the utility calculation, the comparison between these codes is not in the scope of our work. In this framework, which we only defined to evaluate our technique, we use RS codes. A (N, K) Reed Solomon code with K source data symbols and $N - K$ parity symbols is able to protect data against $N - K$ symbol erasures. By applying RS coding vertically in Fig. 4.1, the probability of successfully receiving (PSR) of the i^{th} packet with $RS(N, C_i)$ over a network with packet loss rate of p is:

$$PSR_i = \sum_{k=0}^{N-S_i} \binom{N}{k} p^k \times (1-p)^{N-k}. \tag{4.2}$$

Given a total bit budget (R_{total}), the channel rate allocation technique determines the number of parity bits of each NAL unit (B_{C_i}) in such a way that the total utility is

maximized. The total expected utility (TEU) is calculated as:

$$TEU = \sum_{i=1}^{L} (U_i \times PSR_i), \tag{4.3}$$

where PSR_i is the probability of successfully receiving the i^{th} packet and is calculated as in Eq. (4.2). U_i is the utility of packet i and defined in terms of distortion decrement:

$$U_i \;\; = \;\; TSD_i - TSD_0,$$

where TSD_i denotes the total sequence distortion (TSD) when the i^{th} NAL unit is lost and TSD_0 represents the total sequence distortion when there was no loss. In other words, the utility of each packet is the difference of the total distortion of the sequence when the packet is lost and received. The total bit budget consists of all source bits (R_S) and all parity bits (R_C) which are calculated as:

$$R_S = \sum_{i=1}^{L} B_{S\_i} = \sum_{i=1}^{L} (S_i \times l_i), \tag{4.4}$$

$$R_C = \sum_{i=1}^{L} B_{C\_i} = \sum_{i=1}^{L} (C_i \times l_i), \tag{4.5}$$

where B_{S_i} and B_{C_i} are distributed into S_i and C_i packets with l_i symbols in each packet. Given a total bit budget (R_{total}), the channel rate allocation technique calculates the proper channel allocation vector ($C = [C_1 \;\; C_2 \;\; ... \;\; C_L]$) in such a way that the total utility is maximized:

$$\begin{aligned} & \max_{C} \; TEU \text{ subject to } R_S + R_C \; \leq R_{Total} \\ = \;\; & \max_{C} \sum_{i=1}^{L} (U_i \times PSR_i) \;\; \text{subject to} \;\; \sum_{i=1}^{L} (C_i \times l_i) \;\; \leq R_{Total} - R_S. \end{aligned} \tag{4.6}$$

It should be mentioned that maximizing the total utility would result in a minimizing the total distortion which means higher quality. Furthermore, in order to solve this optimization problem the Genetic Algorithm (GA) was used. Genetic algorithm is a stochastic and

population-based algorithm which has been successfully employed in many research and optimization problems [134]. In this implementation, each individual characterizes a set of channel rates and is a potential solution. In order to evaluate the fitness of each individual in the population, we use Eq. (4.3) while considering the total bit budget constraint of Eq. (4.6). The implementation used in this chapter is not specifically novel, and more information on the GA implementation can be found in [82].

It should be mentioned that the proposed utility calculation technique does not require the probability of error, but in this particular application, and in order to find the optimized rate allocation for all the layers, we need the probability of successfully receiving each packet. By considering the probability of packet loss, which depends on the packetization scheme, the total expected utility is estimated.

4.3 Utility Estimation of the NAL Units

As it was discussed in the previous section, in order to calculate accurately the utility of each NAL unit, performing multiple-time decoding at the encoder side is required. In this technique, each video packet is discarded from the stream, the stream is decoded, and the total distortion is calculated. The difference between the calculated distortion and the original distortion of the stream is considered as the utility of that video packet. Although this calculation would give an accurate calculation of the utility, it requires huge extra computation, which might not be practical in many applications. In this work, we estimate the utility of each NAL unit by using a simple technique at the encode time. In fact, our technique calculates the utility at the MB level and can be used for the case of multiple slices per frame too.

In order to simplify the problem, we start from the single spatial layer case and then extend the estimation to multiple spatial layers. Assuming the i^{th} NAL unit corresponds to frame n, we use $TSD(n)$ as the total sequence distortion when frame n is lost. Also, at the block level, we define $TSD(n,m)$ as the total sequence distortion when block m in frame n is lost. Using these notations, the utility of frame n is calculated as:

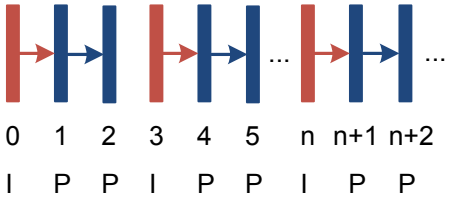
$$\begin{aligned} U(n) &= \sum_{m=1}^{M} U(n,m) \\ &= \sum_{m=1}^{M} \Big(TSD(n,m) - TSD_0 \Big), \end{aligned} \tag{4.7}$$

where $U(n,m)$ is the utility of block m in frame n and M is the total number of blocks in each frame. By assuming there is no error propagation within the frame from neighbouring blocks, Eq. (4.7) defines the utility of frame n as the sum of the utility of all the blocks in that frame. It should be noted that if all the packets are received at the decoder, the total distortion would be equal to total source coding distortion. Also, in the calculation of $TSD(n,m)$, the assumption is that all the slices before and after frame n are received correctly at the decoder. It means that there was no error propagation before frame n. In this section, we will consider different prediction structures. These structures are shown in Fig. 4.2. We start from the simplest case, and move to the prediction structure used in temporal prediction of SVC.

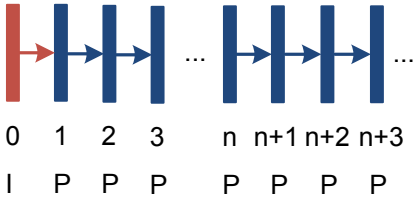
4.3.1 IPPIPP structure

This is the simplest structure among the three structures shown in Fig. 4.2. In this structure, a loss in frame n, assuming that $n = 3k$ for some $k \in \mathbb{N}$, would only propagate to frames $n+1$ and $n+2$. Frame $n+3$ is encoded as an I frame, which stops all the previous error propagations. It should be noted that the utility calculation for frames $n+1$ and $n+2$ are simplified versions of the utility calculation for frame n. So, in this section, we only show the utility calculation for frame n. The total sequence distortion for the error free case is calculated by using Eq. (2.22) and assuming $p = 0$ and $D_{\text{ep}} = 0$, as:

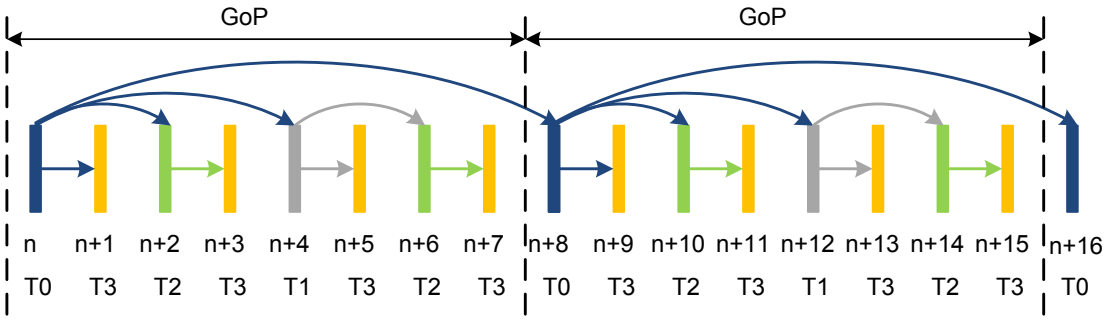
$$\begin{aligned}
TSD_0 &= D(n) + D(n+1) + D(n+2) + \sum_{\substack{i\in\{1\ldots N\}\ \&\\ i\notin\{n,n+1,n+2\}}} D(i) \\
&= D_{\text{src}}(n) + D_{\text{src}}(n+1) + D_{\text{src}}(n+2) + \sum_{\substack{i\in\{1\ldots N\}\ \&\\ i\notin\{n,n+1,n+2\}}} D(i) \\
&= D_{\text{src}}(n,m) + \sum_{\substack{j\in\{1\ldots M\}\\ \&\ j\neq m}} D_{\text{src}}(n,j) + \sum_{j\in L^{n+1}_{(n,m)}} D_{\text{src}}(n+1,j) + \sum_{\substack{j\in\{1\ldots M\}\ \&\\ j\notin L^{n+1}_{(n,m)}}} D_{\text{src}}(n+1,j) \\
&\quad + \sum_{j\in L^{n+2}_{(n,m)}} D_{\text{src}}(n+2,j) + \sum_{\substack{j\in\{1\ldots M\}\ \&\\ j\notin L^{n+2}_{(n,m)}}} D_{\text{src}}(n+2,j) + \sum_{\substack{i\in\{1\ldots N\}\ \&\\ i\notin\{n,n+1,n+2\}}} D(i), \qquad (4.8)
\end{aligned}$$



(a) IPPIPPIPP



(b) IPPP



(c) GoP = 8

Fig. 4.2 Different prediction structures (a) IPPIPPIPP, (b) IPPP, (c) hierarchical prediction structure with zero delay (GoP =8).

where M is the number of blocks in one frame and N is the total number of frames. $D(n)$ and $D_{\text{src}}(n) = \sum_{m=1}^{M} D_{\text{src}}(n,m)$ denote the end-to-end and source coding distortions of frame n, respectively. $D_{\text{src}}(n,m)$ represents the source coding distortion of block m in frame n. $L_{(n,m)}^{n+1}$ is the set of blocks in frame $n+1$ which are using block m in frame n as a reference. In general, $L_{(n,m)}^{n+k}$ is defined as the set of blocks in frame $n+k$ to which the error from block m in frame n is propagated. A visual presentation of $L_{(n,m)}^{n+k}$ is shown in Fig. 4.3.

It should be mentioned that due to the complexity, we do not consider the error propagation between Intra coded blocks within a frame. This can be one potential source of error in our estimation. For the case that block m in frame n is lost, by using Eq. (2.22) and assuming $p = 1$ for frame n, and $p = 0$ for frames $n+1$ and $n+2$, we will have:

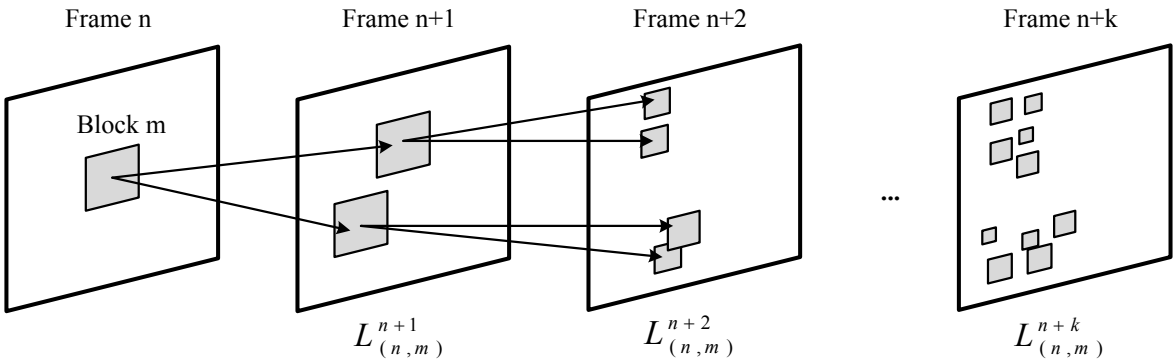


Fig. 4.3 Set of blocks using block m in frame n as a reference.

$$
\begin{aligned}
& TSD(n,m) \\
= \;& D(n) + D(n+1) + D(n+2) + \sum_{\substack{i\in\{1\ldots N\}\ \&\\ i\notin\{n,n+1,n+2\}}} D(i) \\
= \;& D_{\mathrm{ec}}(n,m) + \sum_{\substack{j\in\{1\ldots M\}\\ \&\ j\neq m}} D_{\mathrm{src}}(n,j) + \sum_{j\in L^{n+1}_{(n,m)}} D_{\mathrm{src}}(n+1,j) + \sum_{\substack{j\in\{1\ldots M\}\ \&\\ j\notin L^{n+1}_{(n,m)}}} D_{\mathrm{src}}(n+1,j) \\
& + |L^{n+1}_{(n,m)}|\ D_{\mathrm{ep}}(n,m) + \sum_{\substack{j\in\{1\ldots M\}\ \&\\ j\notin L^{n+1}_{(n,m)}}} D_{\mathrm{ep}}(n,ref(j)) + \sum_{j\in L^{n+2}_{(n,m)}} D_{\mathrm{src}}(n+2,j) + \sum_{\substack{j\in\{1\ldots M\}\ \&\\ j\notin L^{n+2}_{(n,m)}}} D_{\mathrm{src}}(n+2,j) \\
& + \sum_{j\in L^{n+2}_{(n,m)}} D_{\mathrm{ep}}(n+1,ref(j)) + \sum_{\substack{j\in\{1\ldots M\}\ \&\\ j\notin L^{n+2}_{(n,m)}}} D_{\mathrm{ep}}(n+1,ref(j)) + \sum_{\substack{i\in\{1\ldots N\}\ \&\\ i\notin\{n,n+1,n+2\}}} D(i). \qquad (4.9)
\end{aligned}
$$

$D_{\mathrm{ep}}(n,m)$ denotes error propagation distortion and $|L^{n+1}_{(n,m)}|\ D_{\mathrm{ep}}(n,m)$ represents the normalized amount of error propagation from block m of frame n to blocks in $L^{n+1}_{(n,m)}$. For the blocks which are using part of block m as their reference, a ratio of the referred area over the area of the block is used as a weight to $D_{\mathrm{ep}}(n,m)$. Fig. 4.4 shows an example that only a part (with an area of a_1) of block m (with an area of A) is used as reference. In this case the weight is equal to a_1/A.

Furthermore, $ref(j)$ denotes the reference block of block j. Since loss only happened

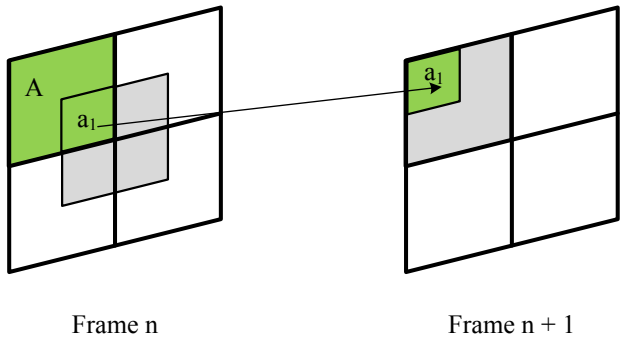


Fig. 4.4 An example of using a part of a block as a reference for the future frame.

in block m of frame n and there was no loss before frame n, we have:

$$\begin{aligned} & D_{\text{ep}}(n-1,j)=0, \\ & \sum_{\substack{j\in\{1\ldots M\}\ \&\\ j\notin L^{n+1}_{(n,m)}}} D_{\text{ep}}(n,ref(j))=0, \\ & \sum_{\substack{j\in\{1\ldots M\}\ \&\\ j\notin L^{n+2}_{(n,m)}}} D_{\text{ep}}(n+1,ref(j))=0. \end{aligned}$$

In addition, by using Eq. (2.24) and assuming $p=1$, we will have:

$$\begin{aligned} D_{\text{ep}}(n,m) &= D_{\text{ec\_rec}}(n,m)+D_{\text{ep}}(n-1,m) \\ &= D_{\text{ec\_rec}}(n,m). \end{aligned} \tag{4.10}$$

where $D_{\text{ec\_rec}}(n,m)$ is the error concealment-reconstructed distortion and is calculated as the sum square difference of reconstructed block and error concealed block. Also, by using Eq. (2.24) and using $p=1$ for frame n, and $p=0$ for frames $n+1$ and $n+2$, we will have:

$$\begin{aligned} \sum_{j\in L^{n+2}_{(n,m)}} D_{\text{ep}}(n+1,ref(j)) &= \sum_{j\in L^{n+2}_{(n,m)}} D_{\text{ep}}(n,m) \\ &= \sum_{j\in L^{n+2}_{(n,m)}} D_{\text{ec\_rec}}(n,m). \end{aligned} \tag{4.11}$$

$\sum_{j \in L^{n+2}_{(n,m)}} D_{\text{ec\_rec}}(n,m)$ represents the normalized amount of error propagation from block m of frame n to blocks in $L^{n+2}_{(n,m)}$ in frame $n+2$. By inserting Eq. (4.10) and Eq. (4.11) in Eq. (4.9), we will have:

$$\begin{aligned} TSD(n,m) &= D(n) + D(n+1) + D(n+2) + \sum_{\substack{i \in \{1 \ldots N\} \ \& \\ i \notin \{n, n+1, n+2\}}} D(i) \\ &= D_{\text{ec}}(n,m) + \sum_{\substack{j \in \{1 \ldots M\} \\ \& \ j \neq m}} D_{\text{src}}(n,j) + \sum_{j \in L^{n+1}_{(n,m)}} D_{\text{src}}(n+1,j) + |L^{n+1}_{(n,m)}| \; D_{\text{ec\_rec}}(n,m) \\ &\quad + \sum_{\substack{j \in \{1 \ldots M\} \ \& \\ j \notin L^{n+1}_{(n,m)}}} D_{\text{src}}(n+1,j) + \sum_{j \in L^{n+2}_{(n,m)}} D_{\text{src}}(n+2,j) + |L^{n+2}_{(n,m)}| \; D_{\text{ec\_rec}}(n,m) \\ &\quad + \sum_{\substack{j \in \{1 \ldots M\} \ \& \\ j \notin L^{n+2}_{(n,m)}}} D_{\text{src}}(n+2,j) + \sum_{\substack{i \in \{1 \ldots N\} \ \& \\ i \notin \{n, n+1, n+2\}}} D(i). \end{aligned} \tag{4.12}$$

Using Eq. (4.8) and Eq. (4.12) and cancelling out equal terms, we will have:

$$\begin{aligned} U(n,m) &= TSD(n,m) - TSD_0 \\ &= D_{\text{ec}}(n,m) + |L^{n+1}_{(n,m)}| \; D_{\text{ec\_rec}}(n,m) + |L^{n+2}_{(n,m)}| \; D_{\text{ec\_rec}}(n,m) - D_{\text{src}}(n,m). \end{aligned} \tag{4.13}$$

For frame n, the total distortion difference is equal to:

$$U(n) = \sum_{m=1}^{M} \Big(D_{\text{ec}}(n,m) + |L^{n+1}_{(n,m)}| \; D_{\text{ec\_rec}}(n,m) + |L^{n+2}_{(n,m)}| \; D_{\text{ec\_rec}}(n,m) - D_{\text{src}}(n,m) \Big). \tag{4.14}$$

$\sum_{m=1}^{M} |L^{n+1}_{(n,m)}| \; \Big(D_{\text{ec\_rec}}(n,m) \Big)$ and $\sum_{m=1}^{M} |L^{n+2}_{(n,m)}| \; \Big(D_{\text{ec\_rec}}(n,m) \Big)$ represent the total error concealment-reconstructed propagation from frame n to frame $n+1$ and frame $n+2$ respectively. We would refer to this case as two levels of error propagation, where the first level is to frame $n+1$ and the second level is to frame $n+2$.

4.3.2 IPPP structure

In previous case, error from frame n would only propagate to frame $n+1$ and $n+2$, where in the IPPP case (Fig. 4.2-b), error from frame n can propagate to more frames in future. Assuming $N-n$ denotes the number of frames that error would propagate from n, and based on Eq. (4.14), we will have:

$$\begin{aligned} U(n) &= \sum_{m=1}^{M} \Big(TSD(n,m) - TSD_0 \Big) \\ &= \sum_{m=1}^{M} \Big(D_{\text{ec}}(n,m) - D_{\text{src}}(n,m) \Big) + \sum_{m=1}^{M} \sum_{i=1}^{N-n} |L_{(n,m)}^{n+i}| \; D_{\text{ec\_rec}}(n,m). \end{aligned} \tag{4.15}$$

4.3.3 Hierarchical prediction structure with zero delay

The concept of multiple references of H.264/AVC can lead to other prediction structures. Fig. 4.2-c is an example of hierarchical prediction structure. The numbers below the pictures show the coding order and Tk denotes the k^{th} temporal layers. The reference picture list, list0, of a picture with temporal layer identifier k is restricted to pictures with temporal layer identifiers less than k. As a result, the coded picture can be decoded without help of pictures with temporal layer identifiers greater than k. This structure has a delay of zero pictures and provides four temporal layers. Based on this structure, each frame might be used for prediction of more than one frames. For example, frame n is used as a reference for coding of frames $n+1$, $n+2$, $n+4$ and $n+9$. Also, frames in the last temporal layers (frames $n+1$, $n+3$, $n+5$ and $n+7$) are not used for prediction of future frames. As a result, errors occurring in these frames will not be propagated to other frames. In addition, the level of error propagation is different for each temporal layers. For GoP size of 8, the levels of error propagation for temporal layer 3 to 1 are 0, 1 and 2, respectively. For frames in temporal layer zero, the level of error propagation would be upto $Number\ of\ GoPs - 1 + 3$. If only error propagation within the current GoP is considered, the number would be 3. Based on Eq. (4.15), we will have:

$$\begin{aligned} U(n) &= \sum_{m=1}^{M} \Big(TSD(n,m) - TSD_0 \Big) \\ &= \sum_{m=1}^{M} \Big(D_{\text{ec}}(n,m) - D_{\text{src}}(n,m) \Big) \end{aligned}$$

$$+\sum_{m=1}^{M}\sum_{i=1}^{L}\sum_{k \text{ in } \{\text{frames in } i\text{-level of propagation from } n\}} |L^{k}_{(n,m)}|\ D_{\text{ec\_rec}}(n,m). \tag{4.16}$$

For instance, assuming the GoP structure shown in Fig. 4.2-c, for frame $n+4$, $L=2$ and we have:

$$\begin{aligned}
U(n+4) &= \sum_{m=1}^{M}\Big(D_{\text{ec}}(n+4,m) - D_{\text{src}}(n+4,m)\Big) \\
&\quad + \sum_{m=1}^{M}\sum_{i=1}^{2}\sum_{k \text{ in } \{\text{frames in } i\text{-level of propagation from } n+4\}} |L^{k}_{(n+4,m)}|\ D_{\text{ec_rec}}(n+4,m) \\
&= \sum_{m=1}^{M}\Big(D_{\text{ec}}(n+4,m) - D_{\text{src}}(n+4,m)\Big) \\
&\quad + \sum_{m=1}^{M}|L^{n+5}_{(n+4,m)}|D_{\text{ec_rec}}(n+4,m) + \sum_{m=1}^{M}|L^{n+6}_{(n+4,m)}|D_{\text{ec_rec}}(n+4,m) \\
&\quad + \sum_{m=1}^{M}|L^{n+7}_{(n+4,m)}|D_{\text{ec_rec}}(n+4,m).
\end{aligned}$$

4.3.4 Spatial scalability

Since there is no error propagation from higher spatial layers to the lower layers, the utility calculation in the highest enhancement spatial layer is done in the same way as the single spatial layer, which was explained previously. For lower spatial layers, error propagation to higher spatial layers should also be considered. Assuming a loss happened spatial layer s of frame n, the error might propagate to frame $n+1$ spatial layer s, frame n spatial layer $s+1$ and frame $n+1$ spatial layer $s+1$ and etc.. It should be mentioned that error propagation from frame n spatial layer s, to frame $n+1$ spatial layer $s+1$ may happen in two paths:

- Frame n layer $s \rightarrow$ frame n layer $s+1 \rightarrow$ frame $n+1$ layer $s+1$
- Frame n layer $s \rightarrow$ frame $n+1$ layer $s \rightarrow$ frame $n+1$ layer $s+1$

Based on the explanation, in order to calculate the utility of frame n, spatial layer s, Eq. (4.16) would be modified to:

$$U^s(n) = \sum_{m=1}^{M} \left(D^S_{\text{ec}}(n,m) - D^S_{\text{src}}(n,m) \right) + \sum_{m=1}^{M} \sum_{i=1}^{L} \sum_{j=s}^{S} \sum_{k \text{ in } \{\text{frames in } j\text{-th spatial layer and } i\text{-level of propagation from } n\}} |L^k_{(n,m)}| \; D_{\text{ec\_rec}}(n,m). \quad (4.17)$$

where S denotes the number of spatial layers and $D^S_{\text{ec}}(n,m)$, $D^S_{\text{src}}(n,m)$ represent error concealment and source distortions in the highest spatial layer. It should be mentioned that since PSNR calculation is usually done at the highest spatial layer, our distortion calculation should be performed at the highest spatial layer. Also, it is assumed that each lost slice or picture is concealed by copying from previous picture in the same spatial layer. In addition, if a lost happen in the lower spatial layer, the higher spatial layer are assumed to be lost too.

4.4 Simulation Results

The proposed technique was implemented in the JSVM software. The source distortion (D_{src}), error concealment-reconstructed distortion ($D_{\text{ec\_rec}}$) and original error concealment distortion (D_{ec}) were calculated at the encode time and stored per block. Based on the error propagation and prediction structure, the utility is calculated. It should be mentioned that our proposed method does not require multiple-pass decoding at the encoder side. To study the performance of the proposed technique, a set of simulations were conducted with conditions mentioned in Section 3.4 and with the following differences:

- JSVM 9.15 was used as the SVC encoder and decoder.
- Each sequence was coded with five temporal, and two spatial layers. 25% of the total bit rate is spent for coding of the base spatial layer, and the 75% is allocated to the enhancement spatial layer. In each spatial layer, the initial QPs of the lowest to the highest temporal layers changes by -2, 1, 3 and 4 respectively.
- Standard sequences, “Football”, “Foreman”, “Mobile” and “News” were encoded

with QCIF and CIF sizes, and "City", "Crew", "Ice", "Harbour" were coded with CIF and 4CIF sizes for the base and enhancement spatial layers respectively. The streams were coded at 30 fps.

- The "FixedQPEncoderStatic" tool in JSVM was used to encode the video with fixed bit rates. This tool finds the appropriate QP for the target bit rate. The QCIF-CIF sequences were coded at 192, 375, 750 and 1500 kbps and the bit rates used for coding the CIF-4CIF sequences were 375, 750, 1500 and 3000 kbps.

- Each frame was divided into three slices. Each packet includes one slice. Based on the assumption that RTP/UDP/IP transmission is used, lost or damaged packets are discarded without retransmission.

- The extra channel rate allocated for protecting the coded stream is equal to 25% of the source rate ($R_C = 0.25 \times R_S$).

- Sequence parameter set (SPS) and picture parameters set (PPS), which are essential for decoding the stream, are protected with highest priority.

- In order to describe a network with losses, we used a two-state Markov model which is usually referred as Gilbert model [135]. The reason for using a Gilbert model instead of the network simulator used in simulations of chapter 3, was to consider bursty behaviour of the channel in addition to loss probability. The average packet loss rates (PLR) of 0%, 5%, 10%, 15% and 20% with average burst error length of 2 were used in the simulations.

- Due to random nature of the channel and in order to get consistent results, each transmission was repeated with 200 channel realization and the average peak signal to noise ratio (PSNR) was calculated.

- A simple Picture Copy (PC) was used at the decoder side to conceal the lost packets.

4.4.1 Estimation accuracy

In order to estimate the accuracy of our proposed method, the estimated utilities are compared to the utility calculated based on the multiple decoding at NAL unit level. As it was mentioned in Section 4.1, this technique give us the most accurate utility for each NAL

unit. In the rest of this paper, we refer to this technique as "Actual" utility calculation technique. We studied the accuracy of our estimation for all prediction structures shown in Fig. 4.2. In particular, we did the experiments for hierarchical prediction structure similar to Fig. 4.2-c and GoP size of 16. Since this case includes all the previous structures, we only report the results for this prediction structure. In this prediction structure, the levels of error propagation for temporal layer 4 to 1 are 0, 1, 2 and 4, respectively. For frames in the base temporal layer, the level of error propagation would be up to $Number\ of\ GoPs-1+4$. If only error propagation within the current GoP is considered, the number would be 4.

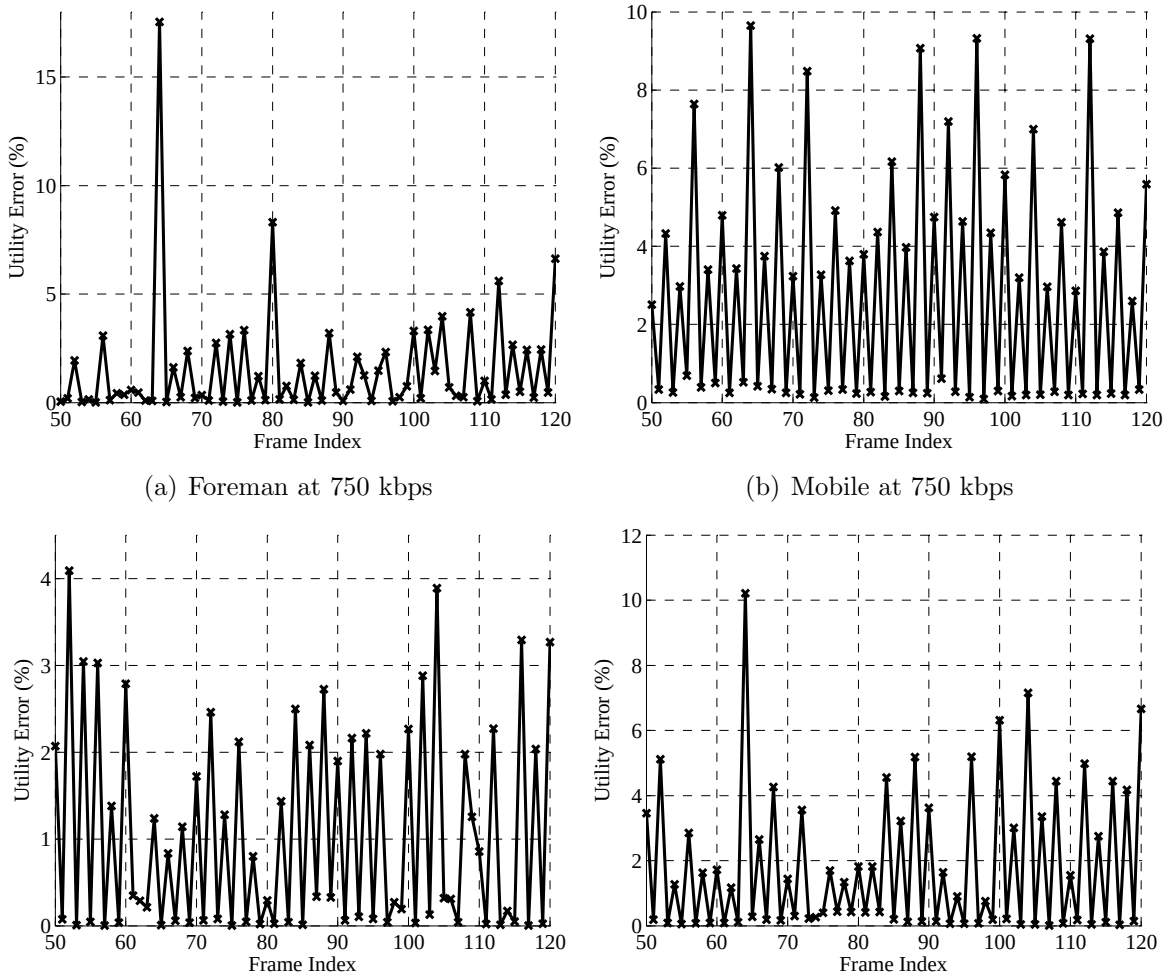
The estimation error of our technique for frame 50 to 120 of "Foreman" and "Mobile" sequences coded at 750 kbps and "City" and "Crew" sequences coded at 1500 kbps are show in Fig. 4.5. The estimation error is calculated as:

$$\text{Estimation error} \quad = \quad \frac{\Big|\text{Actual utility} - \text{Estimated utility}\Big|}{\text{Actual utility}} \times 100 \tag{4.18}$$

It can be observed that, in highest temporal layer, which are represented in odd frame indexes, the estimation error is very low. These frames are not used as a reference and consequently, there is no error propagation from these frames to future frames. In fact, the utility of these frames is calculated as:

$$U(n) \quad = \quad \sum_{m=1}^{M} \Big(D_{\text{ec}}(n,m) - D_{\text{src}}(n,m) \Big).$$

In lower temporal layers, the estimation error is higher. The main reason for the estimation error is that the utility calculation and error propagation are performed at a block level. Estimation at the pixel level achieves the most accurate estimation, but it requires huge computation and storage. Although we are using 4x4 blocks for calculation and storing propagation distortion to future frames, estimation error is introduced because of averaging over 16 pixels. Also, another source of error is in the cases that only a part of a block is used as a reference and a ratio of the referred area over the area of the block is used as a weight for distortion calculation. In addition, error propagation between Intra blocks within a frame, which is not considered due to the complexity, is another potential source



(a) Foreman at 750 kbps

(b) Mobile at 750 kbps

(c) Crew at 1500 kbps

(d) City at 1500 kbps

Fig. 4.5 Utility estimation error vs frame index for (a) "Foreman" sequence (b) "Mobile" sequence with QCIF and CIF sizes at 750 kbps and (c) "Crew" sequence, (d) "City" sequence with CIF and 4CIF sizes at 1500 kbps.

of error. In our simulations, since all the blocks within a slice are in one video packet, all of them are either received or lost. As a result, there will not be any error propagation between two neighbouring Intra blocks with in one slice. In other words, either both are lost and the error concealment technique is used, or they are both received correctly. Furthermore, it should be mentioned that two Intra blocks along the border of two slices are independent. Consequently, although error propagation between two neighbouring Intra blocks is a potential source of the estimation error, it does not introduce any estimation

error in our simulations.

Table 4.1 The average and standard deviation of estimation error for the proposed technique and the proposed low delay technique.

Sequence	Proposed technique		Low delay technique	
	Average (%)	Standard deviation (%)	Average (%)	Standard deviation (%)
Football	1.78	3.09	2.20	5.23
Foreman	2.42	3.31	4.31	14.71
News	4.35	7.04	6.05	14.96
Mobile	2.31	2.63	4.35	14.43
City	1.77	4.78	3.76	13.10
Crew	1.66	2.65	2.93	7.89
Harbour	3.47	6.33	5.30	13.57
Ice	1.55	4.92	3.08	11.20
Average	2.41	4.34	4.00	11.89

In Table 4.1, the average and standard deviation of the estimation error are shown for different sequences. The average is taken over all the frames and four different bit rates. It can be observed that moving from slow sequences to fast sequences ("News" to "Football"), the accuracy of the estimation increases. This is because of having more Intra blocks and less temporal prediction in faster sequences. In a very slow sequence like "News", most of the areas are not changing or moving. As a result, Inter prediction is used for most of these blocks and the temporal prediction might continue for multiple GoPs. In addition, it was observed that by increasing the bit rate, the accuracy of our estimation improves. It is because of the fact that by increasing the bit rate, more blocks are coded as Intra and consequently, less temporal prediction is used.

It can be observed that by using the proposed technique the utility of each NAL unit can be estimated very accurately, and the average error is 2.41%. Although the computational complexity of the proposed technique is much lower than actual utility calculation, it still has an introduced delay problem. Since the utility calculator needs to consider the error propagation to future frames, it is required to wait until they are encoded. This delay is not tolerable in some real time applications. In order to address this issue, we studied the results and the prediction structures deeply and realized that we can preserve the good estimation accuracy by considering limited levels of propagation to future frames.

Table 4.2 The average utility per area for different temporal layers at 750 kbps.

Sequence	Temporal layer 0	Temporal layer 1	Temporal layer 2	Temporal layer 3	Temporal layer 4
Football	28.05	18.81	11.63	5.35	2.15
Foreman	145.38	17.50	5.25	1.58	0.32
News	22.63	4.87	2.19	0.59	0.11
Mobile	421.93	60.23	23.49	7.20	1.37
City	103.80	20.26	8.87	3.36	1.26
Crew	29.64	8.40	3.56	1.35	0.42
Harbour	186.59	28.53	8.64	3.12	0.84
Ice	229.11	33.08	12.01	3.51	0.85

In Table 4.2, the average utilities calculated per temporal layer are shown for different sequences at 750 kbps. The utilities are normalized based on the resolution of the video. It can be observed that the utility would decrease by moving to higher temporal layers. This is because of the used prediction structure and the higher potential levels of error propagation in lower temporal layers. Also, the utility of the base temporal layer is significantly higher than other temporal layers. The reason is that an error in the base temporal layer frames can propagate to all the frames in current GoP and the frames in future GoPs. In our low delay technique, the error propagation is only considered in the current GoP and the maximum delay is equal to the size of one GoP. Based on the hierarchical prediction structure (Fig. 4.2-c), the utilities of NAL units in temporal enhancement layers are calculated in the same way. The only difference would be in the base layer. The estimated utilities for these NAL units will be less than the actual utility. But, it should be considered that, even by considering limited levels of propagations for these NAL units, their utilities are higher than the utility of enhancement layers. As a result, the rate allocation techniques would protect them with more channel bits. By using the low delay proposed technique, and by assuming a GoP size of 16 and coding 30 frames per second, the introduced delay would be less than 0.5 sec which can be easily handled by buffering.

The average and standard deviation of the estimation error of the proposed low delay technique are shown in Table 4.1. Compared to our full level estimation technique, the estimation error has higher average and standard deviation. These numbers are higher in slow sequences like “News”. As it was mentioned, in slow sequences more blocks are coded

Table 4.3 The average and standard deviation of estimation error in the base temporal layer for the proposed technique and the proposed low delay technique.

Sequence	Proposed technique		Low delay technique	
	Average (%)	Standard deviation (%)	Average (%)	Standard deviation (%)
Football	4.68	5.72	11.58	15.14
Foreman	20.85	20.42	51.39	33.56
News	13.73	16.05	41.15	30.07
Mobile	17.10	19.11	50.04	33.50
City	13.45	14.28	45.41	30.08
Crew	4.36	5.27	24.74	18.84
Harbour	14.70	15.92	44.22	30.37
Ice	13.83	14.28	38.62	26.14
Average	12.84	13.88	38.39	27.21

as Inter or Skip and a transmission error might propagate for a few GoPs. Since we don't consider the error propagation beyond a GoP, the accuracy of the estimated utility of base temporal layer is lower. Table 4.3 illustrates the average and standard deviation of the estimation error in the base temporal layer for the proposed techniques. Comparing to Table 4.1, it can be observed that the estimation error in the base temporal layer is higher than the average over all frames. In addition, and as it was expected, the estimation error in the low delay technique has higher average and standard deviation in the base temporal layer.

In order to compare the computational complexity of the proposed method with the actual utility calculation, we measured the times for each method. All the simulations are conducted on a PC with Intel Core i7-2600 Processor (8M Cache, 3.40 GHz) and 8 GB of RAM. The measured times are the sum of the SVC encoding time and the utility calculation time. On average, over all sequences and bit rates, the execution time of the encoding with our utility measurement technique represents only 17.6% of the time necessary for encoding with utility calculation through multiple decodings, i.e. a speed up by a factor of nearly 6. Moreover, in our technique, the utility calculation only increased the encoding time by 25.4% as compared to a standard SVC encoder that does not perform any utility calculation.

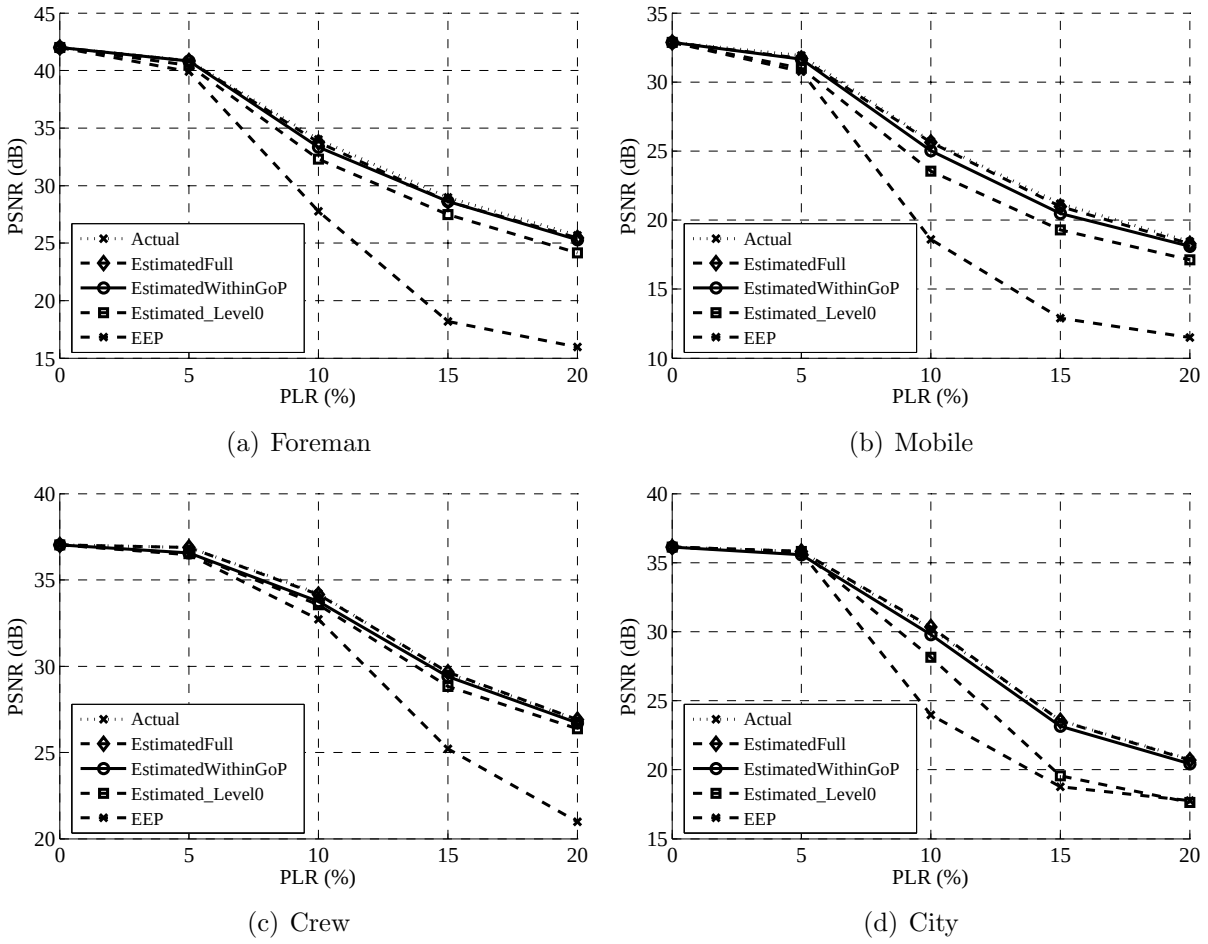
4.4.2 Video coding quality

In this section, the proposed techniques are studied in terms of the quality of the received video at the decoder side. In Fig. 4.6, average PSNR vs PLR curves of different methods are shown for "Foreman" and "Mobile" with QCIF and CIF sizes at 750 kbps and for "City" and "Crew" sequences with CIF and 4CIF sizes coded at 1500 kbps. "Actual" represents the actual utility calculation by using multiple decoding per NAL unit with very high complexity. "EstimatedFull" and "EstimatedWithinGoP" respectively denote our proposed technique and the low delay version which does not consider propagation beyond the current GoP. In "Estimated_level0" method, the utility of each NAL unit is calculated by considering only zero level of propagation. In other words, this technique does not use error propagation to future frames and has zero delay. For this technique, the utility calculation in Eq. (4.17) is modified to:

$$U^s(n) \quad = \quad \sum_{m=1}^{M} \Big(D^S_{\text{ec}}(n,m) - D^S_{\text{src}}(n,m) \Big). \tag{4.19}$$

Furthermore, equal error protection (EEP) protects all the NAL units equally. In order to make the comparisons fair, we used the utility of the "Actual" technique in order to adjust the genetic algorithm parameters and used the same parameters for all other UEP techniques. In this way, the results would only reflect the effect of different utility estimation techniques. We set the population size to 500 and the number of generations to 2000. We also set the crossover and mutation probabilities to 0.7 and 0.1 respectively.

It can be observed that "Actual" and "EstimatedFull" techniques have the best performances. These two curves are very close and can hardly be differentiated. Using our low delay estimation technique ("EstimatedWithinGoP"), the performance would slightly decrease. The "Estimated_level0' method has lower PSNR because of the inaccurate utility estimation and "EEP" has lower performance compared to the UEP techniques. By increasing the PLRs the performance difference of the techniques would be more significant. In low PLRs, since the number of parity bits for protection of each NAL unit is lower, the added channel bits are enough to protect all the packets. Consequently, the curves are very close in low packet loss rates. By increasing the PLR, it would be critical to allocate the limited channel rate to more important units and the accuracy of the utility estimation has a major role.

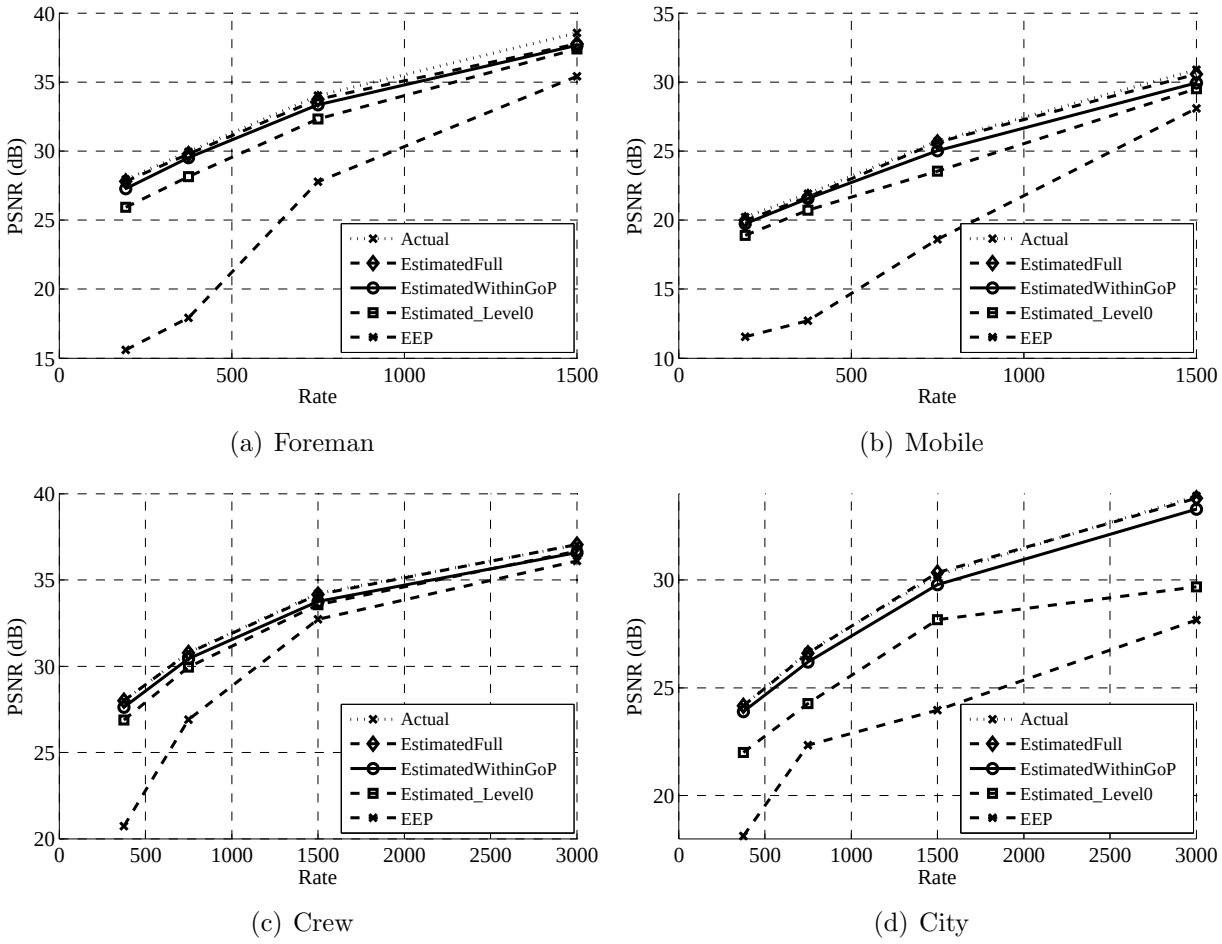


(a) Foreman (b) Mobile

(c) Crew (d) City

Fig. 4.6 PSNR vs PLR of different methods for (a) "Foreman" sequence (b) "Mobile" sequence with QCIF and CIF sizes at 750 kbps and (c) "Crew" sequence, (d) "City" sequence with CIF and 4CIF sizes at 1500 kbps.

The performance curves of these techniques at different bit rates are illustrated in Fig. 4.7. The videos are coded with two spatial layers with QCIF and CIF size for "Foreman" and "Mobile" sequences and CIF and 4CIF sizes for "City" and "Crew" sequences. The protected streams are transmitted over channels with 10% packet loss rate and average burst error length of 2. It can be noticed that "Actual" and "EstimatedFull" outperform other techniques at different bit rates. "EstimatedWithinGoP" has slightly lower average PSNR and "EEP" achieves the lowest performance. In order to study the performance of each technique deeply, the average PSNR difference of each method compared to "Actual", which is our reference, is shown in Table 4.4 for all the tested sequences. The average is



(a) Foreman (b) Mobile

(c) Crew (d) City

Fig. 4.7 PSNR vs Rate of different methods for (a) "Foreman" sequence (b) "Mobile" sequence with QCIF and CIF sizes and (c) "Crew" sequence, (d) "City" sequence with CIF and 4CIF sizes. Tranmisted over a channel with 10% packet loss rate and average burst error length of 2.

taken over four different bit rates, and four packet loss rates (5%, 10%, 15% and 20%). Using EEP method, the average PSNR is 5.92 dB lower than using the high complexity "Actual" utility estimation technique. As we expected based on the estimation accuracy of our proposed technique, our full level estimation technique would achieve almost the same performance as the "Actual" technique with much lower complexity. In application which cannot tolerate the delay, our low delay version with slightly lower performance (0.23 dB) can be employed. Also, the zero delay utility estimation "Estimated_level0" technique has a performance loss of 1.10 dB.

Table 4.4 The average delta PSNR of each technique compared to the "Actual" method.

Sequence	EEP (dB)	Estimated level0 (dB)	Estimated within GoP (dB)	Estimated full (dB)
Football	5.16	0.61	0.02	0.00
Foreman	8.28	1.34	0.38	0.19
News	8.42	0.68	0.11	0.01
Mobile	6.39	1.44	0.39	0.11
City	4.43	2.29	0.34	0.02
Crew	4.06	0.70	0.31	0.01
Harbour	4.50	1.10	0.15	0.05
Ice	6.09	0.63	0.13	0.05
Average	5.92	1.10	0.23	0.05

In order to provide a better representation, a subjective comparison of different techniques is illustrated in Fig. 4.8. The images represent the decoded frame number 71 of Foreman sequence. The sequences were encoded at 750 kbps with CIF size and 30 fps. Encoded videos were transmitted over a channel with 10% packet loss and same loss pattern was used for all the five techniques. Five temporal and two spatial layers are included in each stream. The quality of the images for the proposed techniques are very close to the quality of the actual utility calculation technique.

4.5 Chapter Summary

In many multimedia applications, coded video is transmitted over error prone heterogeneous networks. Because of the predictive mechanism used in video coding, transmission error would propagate temporally and spatially and would result in significant quality losses. In order to address this problem, different error resilience methods have been proposed. One of the techniques, which is commonly used in video streaming, is unequal error protection (UEP) of scalable coded video. In this technique, different independent layers of an SVC stream are protected differently and based on their importance by using forward error correction (FEC) codes. Accurately analysing the importance or utility of each video part is a critical component and would lead to a better protection and higher quality of the received video. Calculation of the utility is usually based on multiple decoding of sub bitsreams and

is highly computationally complex. In this work, we proposed an accurate low complexity utility estimation technique that can be used in different applications. This technique estimates the utility of each network abstraction layer (NAL) by considering the error propagation to future frames. We utilized this method in order to do UEP on the scalable extension of H.264/AVC codec and it achieved almost the same performance as highly complex estimation techniques (an average loss of 0.05 dB). Furthermore, we proposed a low delay version of this technique that can be used in delay constraint application. The estimation accuracy and performance of our proposed technique were studied extensively.

(a) Actual

(b) Estimated full

(c) Estimated with GoP

(d) Estimated level0

(e) EEP

Fig. 4.8 Subjective results for "Foreman" sequence frame 71 with CIF size and 30 fps at 750 kbps, packet loss rate of 10%.

Chapter 5

Conclusion

5.1 Research Contributions

In many multimedia applications, coded video is usually transmitted over unreliable channels. Due to predictive video coding, channel errors propagate spatially and temporally and may decrease the quality of the received video significantly. In this thesis, we described a number of contributions in order to address the problem of video transmission over unreliable channels. Our main focus was on forward error correction techniques which add redundancy at the encoder side to make the bitstream more resilient. We used scalable video streams coded by using the scalable extension of H.264/AVC. However, the proposed techniques can be used with any other scalable video encoder. It should be noted that the proposed techniques can be employed individually or used with other error resilience schemes to achieve better performance. The main research achievements of this dissertation are:

- We proposed two prediction structures that form more robust reference frames in order to reduce the introduced mismatch between the encoder and the decoder. Generally, modifying the conventional prediction structure is an approach to reduce the propagation of error to succeeding frames. In this approach, instead of using the current reconstructed frame as a reference for the future frame, a less error vulnerable modified reconstructed frame is employed. Our proposed techniques combine error robustness of previous Intra coded blocks, better prediction achieved by using the previous reference frame, and exponential decay of error propagation caused by the

leaky prediction. Simulation results showed the effectiveness of our scheme, especially for medium and high motion sequences.

- Since modifying the prediction structure changes the end-to-end distortion estimation, we calculated the end-to-end distortion of the second proposed prediction structure based on the LARDO technique [28]. By calculating the end-to-end distortion, we used our proposed prediction structure in combination with error resilience mode decision. Thus, we achieved the idea of having more robust reference frame with selecting the best mode by considering the trade off between compression efficiency and error robustness. By using the estimated end-to-end distortion in the mode decision process, we improved the performance of our technique by 2.7 dB on average. It should be mentioned that the end-to-end distortion estimation requires an estimate of the channel packet loss rate which might not be available in some applications.

- We proposed an accurate low complexity utility estimation technique that can be used in unequal error protection of scalable coded video. Applying unequal error protection on scalable coded video is employed in many multimedia applications. In this technique, different layers of an SVC stream are protected differently and based on their utility by using forward error correction codes. In the proposed technique, the utility of each NAL unit is calculated by considering the source coding distortion, error concealment distortion and the error propagation to future frames. A low delay version of this technique was also presented that can be employed in delay constraint application. The estimation accuracy and performance of our proposed technique were studied extensively.

5.2 Future Work

In previous chapters, we described different error resilience video coding techniques. In this section, we present a few topics that we believe can continue this research.

- The estimated end-to-end distortion that we proposed in Chapter 3, can be used in error resilience motion estimation. By using this distortion, motion vectors referring to safer areas would be selected. Using safer areas as reference results in less temporal error propagation.

- As it was mentioned, the end-to-end distortion estimation in Chapter 3 requires an estimate of the channel packet loss rate which might not be available in some applications. The impact of inaccurate packet loss estimation on the performance of this technique could be the topic of future studies.

- The utility calculation approach that we proposed in Chapter 4 was only tested in a very simple framework. Other parts of this framework such as rate allocation technique, packetization and parity coders can be improved to achieve higher overall performance. Also, the calculated utility can be employed in congestion control algorithms that require to prioritize the video packets.

- In the proposed end-to-end distortion estimation and similar techniques [25, 26, 28], it is assumed that constraint Intra prediction is used. Constraint Intra prediction means Intra prediction is only done by using the neighbouring Intra coded pixels. Although constrained Intra prediction helps error robustness, it decreases the coding efficiency especially in lower bit rates. In order to use unconstrained Intra prediction, the proposed end-to-end distortion estimation should be modified.

References

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

[2] 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, Sep. 2007.

[3] H. Schwarz, D. Marpe, T. Schierl, and T. Wiegand, "Combined scalability support for the scalable extension of H.264/AVC," in *Proceedings of IEEE International Conference on Multimedia and Expo, ICME*, Amsterdam, The Netherlands, Jul. 2005.

[4] I. E. G. Richardson, *H.264 and MPEG-4 video compression: video coding for next-generation multimedia*. John Wiley Sons Ltd., 2003.

[5] M. Ghanbari, *Standard Codecs: image compression to advanced video coding*. London, UK: IEE Telecommunications Series, 2003.

[6] *Video Codec for Audiovisual Services at p x 64 kbit/s*. ITU-T Recommendation H.261, ITU-T, Nov. 1990.

[7] *Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to About 1.5 Mbit/sPart 2: Video*. ISO/IEC 11172-2 (MPEG-1 Video), ISO/IEC JTC 1, Mar. 1993.

[8] *Generic Coding of Moving Pictures and Associated Audio Information Part 2: Video*. ITU-T Recommendation H.262 and ISO/IEC 13818-2 (MPEG-2 Video), ITU-T and ISO/IEC JTC 1, May 1996.

[9] *Video Coding for Low Bit Rate communication*. ITU-T Recommendation H.263, Nov. 1995.

[10] *Coding of audio-visual objectsPart 2: Visual*. ISO/IEC 14492-2 (MPEG-4 Visual), ISO/IEC JTC 1, Apr. 1999.

[11] *Advanced Video Coding for Generic Audiovisual Services.* ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), May 2003.

[12] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," in *RFC 1889*, Jan. 1996.

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

[14] J. V. T. of ITU-T VCEG and I. MPEG, "Scalable video coding," JVT-N020, Jan. 2005.

[15] G. Cote and F. Kossentini, "Optimal Intra coding of blocks for robust video communication over the internet," *Signal Processing: Image Communications*, vol. 15, no. 1, pp. 25–34, Sep. 1999.

[16] Q. F. Zhu and L. Kerofsky, "Joint source coding, transport processing and error conncealment for H.323-based packet video," in *Proceeding of Society of Photographic Instrumentation Engineers (SPIE)*, San Jose, USA, Jan. 1999, pp. 56–62.

[17] R. M. Schreier and A. Rothermel, "Motion adaptive Intra refresh for the H.264 video coding standard," *IEEE Transactions on Consumer Electronics*, vol. 52, no. 1, pp. 249–253, 2006.

[18] R. Satyan, S. Nyamweno, and F. Labeau, "Comparison of Intra updating methods for H.264," in *Proceeding of IEEE International Symposium on Wireless Personal Multimedia Communications*, Jaipur, India, Dec. 2007, pp. 996–999.

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

[20] H. J. Ma, F. Zhou, R. X. Jiang, and Y. W. Chen, "A network-aware error-resilient method using prioritized Intra refresh for wireless video communications," *Journal of Zhejiang University*, vol. 10, no. 8, pp. 1169–1176, 2009.

[21] P. Nunes, L. D. Soares, and F. Pereira, "Error resilient macroblock rate control for H.264/AVC video coding," in *Proceedings of the International Conference on Image Processing (ICIP)*, San Diego, California, USA, Oct. 2008, pp. 2132–2135.

[22] P. Nunes, L. D. Soares, and F. Pereira, "Automatic and adaptive network-aware macroblock Intra refresh for error-resilient H.264/AVC video coding," in *Proceedings of the International Conference on Image Processing (ICIP)*, Cairo, Egypt, Nov. 2009, pp. 3073–3076.

[23] S. Nyamweno, R. Satyan, and F. Labeau, "Error resilient video coding via weighted distortion," in *Proceeding of IEEE International Conference on Multimedia Expo*, New York, USA, Jun. 2009, pp. 734–737.

[24] S. Nyamweno, R. Satyan, and F. Labeau, "Intra-distance derived weighted distortion for error resilience," in *Proceedings of the International Conference on Image Processing (ICIP)*, Cairo, Egypt, Nov. 2009, pp. 1057–1060.

[25] T. Stockhammer, T. Wiegand, and D. Kontopodis, "Rate-distortion optimization for JVT/H.26L coding in packet loss environment," in *Proceeding of IEEE International Packet Video Workshop*, Pittsburgh, USA, Apr. 2002.

[26] R. Zhang, S. Regunathan, and K. Rose, "Video coding with optimal Inter/Intra-mode switching for packet loss resilience," *IEEE Journal on Selected Areas Communication*, vol. 18, no. 6, pp. 966–976, 2000.

[27] H. Yang and K. Rose, "Recursive end-to-end distortion estimation with model-based cross-correlation approximation," in *Proceedings of the International Conference on Image Processing (ICIP)*, Barcelona, Spain, Sep. 2003, pp. 469–472.

[28] Y. Zhang, W. Gao, Y. Lu, Q. Huang, and D. Zhao, "Joint source-channel rate-distortion optimization for H.264 video coding over error-prone networks," *IEEE Transactions on Multimedia*, vol. 9, no. 3, pp. 445–454, 2007.

[29] Y. Zhang, W. Gao, H. Sun, Q. Huang, and Y. Lu, "Error resilience video coding in H.264 encoder with potential distortion tracking," in *Proceedings of the International Conference on Image Processing (ICIP)*, Singapore, Oct. 2004, pp. 163–166.

[30] H. Huang, C. Wang, and T. Chiang, "A robust fine granularity scalability using trellis-based predictive leak," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 6, pp. 372–385, 2002.

[31] Z. Li and E. Delp, "Channel-aware rate-distortion optimized leaky motion prediction," in *Proceeding of IEEE International Conference on Image Processing (ICIP)*, Singapore, Oct. 2004, pp. 2079–2082.

[32] Y. Liu, J. Prades-Nebot, P. Salama, and E. J. Delp, "Rate distortion analysis of leaky prediction layered video coding using quantization noise modeling," in *Proceeding of IEEE International Conference on Image Processing (ICIP)*, Singapore, Oct. 2004.

[33] H. Yang and K. Rose, "Generalized source-channel prediction for error resilient video coding," in *Proceeding of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toulouse, France, Oct. 2006, pp. 533–536.

[34] R. Satyan, S. Nyamweno, B. Solak, and F. Labeau, "Error resilience using reference frame modification," in *Workshop on Multimedia Signal Processing*, Cairns, Australia, Sep. 2008, pp. 192–195.

[35] J. Goshi, A. Mohr, R. Ladner, E. Riskin, and A. Lippman, "Unequal loss protection for H.263 compressed video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 3, pp. 412–419, Mar. 2005.

[36] T. Fang and L. Chau, "A novel unequal error protection approach for error resilient video transmission," in *Proceeding of the IEEE International Symposium on Circuits and Systems (ISCAS)*, Kobe, Japan, May 2005, pp. 4022–4025.

[37] B. Barmada, M. M. Ghandi, E. Jones, and M. Ghanbari, "Prioritized transmission of data partitioned H.264 video with hierarchical QAM," *IEEE Signal Processing Letters*, vol. 12, no. 8, pp. 577–580, Aug. 2005.

[38] H. Cai, B. Zeng, G. Shen, Z. Xiong, and S. Li, "Error-resilient unequal error protection of fine granularity scalable video bitstreams," *EURASIP Journal on Applied Signal Processing: Special Issue on Advanced Video Technologies and Applications for H.264/AVC and Beyond*, Jun. 2006.

[39] H. Ha and C. Yim, "Layer-weighted unequal error protection for scalable video coding extension of H.264/AVC," *IEEE Transactions on consumer electronics*, vol. 54, no. 2, pp. 736–744, Jan. 2008.

[40] W. C. Wen, H. F. Hsiao, and J. Y. Yu, "Dynamic FEC distortion optimization for H.264 scalable video streaming," in *Proceeding of the IEEE Workshop on Multimedia Signal Processing (MMSP)*, Chania, Greece, Oct. 2007, pp. 147–150.

[41] B. Zhang, L. Xiang, M. Wien, and J.-R. Ohm, "Optimized channel rate allocation for H.264/AVC scalable video multicast streaming over heterogeneous networks," in *Proceeding of the IEEE International Conference on Image Processing (ICIP)*, Hong Kong, China, Sep. 2010, pp. 2917–2920.

[42] B. Zhang, M. Wien, and J.-R. Ohm, "Estimation of the utilities of the NAL units in H.264/AVC scalable video bitstreams," in *Proceeding of 19th European Signal Processing Conference (EUSIPCO)*, Barcelona, Spain, Aug. 2011, pp. 16–20.

[43] V. K. Goyal, "Multiple description coding: compression meets the network," *IEEE Magazine on Signal Processing*, vol. 18, pp. 74–93, Sep. 2001.

[44] Y. Wang, A. R. Reibman, and S. Lin, "Multiple description coding for video delivery," in *Proceeding of IEEE*, Jan. 2005, pp. 57–70.

[45] T. Tillo, M. Grangetto, and G. Olmo, "Redundant slice optimal allocation for H.264 multiple description coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, pp. 59–70, Jan. 2008.

[46] C.-C. Su, H. H. Chen, J. J. Yao, and P. Huang, "H.264/AVC-based multiple description video coding using dynamic slice groups," *Signal Processing: Image Communication*, vol. 23, no. 9, pp. 677–691, Jul. 2008.

[47] J. Apostolopoulos, "Error-resilient video compression through the use of multiple states," in *Proceeding of IEEE International Conference on Image Processing (ICIP)*, Vancouver, Canada, Sep. 2000, pp. 352–355.

[48] I. Radulovic, Y.-K. Wang, S. Wenger, A. Hallapuro, M. M. Hannuksela, and P. Frossard, "Multiple description H.264 video coding with redundant pictures," in *Proceedings of ACM Multimedia Mobile Video Workshop*, Augsburg, Germany, Sep. 2007, pp. 37–42.

[49] T. Turletti and C. Huitema, "RTP payload format for H.261 video streams," in *IETF RFC 2032*, Oct. 1996.

[50] C. Zhu, "RTP payload format for H.263 video streams," in *IETF RFC 2190*, Mar. 1997.

[51] T. Kinoshita, T. Nakahashi, and M. Maruyama, "Variable bit rate HDTV CODEC with ATM cell loss compensation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 3, pp. 230–237, Jun. 1993.

[52] Y. Takishima, M. Wada, and H. Murakami, "Reversible variable length codes," *IEEE Transactions on Communication*, vol. 43, pp. 158–162, Feb. 1995.

[53] D. W. Redmill and N. G. Kingsbury, "The EREC: an error-resilient technique for coding varible-length blocks of data," *IEEE Transactions on Image Processing*, vol. 5, pp. 565–574, Apr. 1996.

[54] R. Farrugia and C. Debono, *Digital Video, ch. 4 Resilient Digital Video Transmission over Wireless Channels using Pixel-Level Artefact Detection Mechanisms.* Floriano De Rango (Ed.), 2010.

[55] W. J. Chu and J. J. Leou, "Detection and concealment of transmission errors in H.261 images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, pp. 74–84, Feb. 1998.

[56] S. S. Hemami and T. H.-Y. Meng, "Transform coded image reconstruction exploiting interblock correlation," *IEEE Transactions on Image processing*, vol. 4, pp. 1023–1027, Jul. 1995.

[57] Y. Wang, Q. F. Zhu, and L. Shaw, "Maximally smooth image recovery in transform coding," *IEEE Transactions on Image Processing*, vol. 41, pp. 1544–1551, Oct. 1993.

[58] P. Salama, N. Shroff, E. Coyle, and E. Delp, "Error concealment techniques for encoded video streams," in *Proceeding of IEEE International Conference on Image Processing (ICIP)*, Washington, USA, Oct. 1995, pp. 9–12.

[59] Y.-K. Wang, M. Hannuksela, V. Varsa, A. Hourunranta, and M. Gabbouj, "The error concealment feature in the h.26l test model," in *Proceeding of IEEE International Conference on Image Processing (ICIP)*, Rochester, USA, Sep. 2002, pp. 729–732.

[60] H. Gharavi and S. Gao, "Spatial interpolation algorithm for error concealment," in *Proceeding of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Las Vegas, USA, Apr. 2008, pp. 1153–1156.

[61] W. M. Lam, A. R. Reibman, and B. Liu, "Recovery of lost or erroneously received motion vectors," in *Proceeding of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Minneapolis, USA, Apr. 1993, pp. 417–420.

[62] J. Lu, M. L. Lieu, K. B. Letaief, and J. I. Chuang, "Error resilient transmission of H.263 coded video over mobile networks," in *Proceeding of IEEE International Symposium on Circuits and Systems*, California USA, Jun. 1998, pp. 502–505.

[63] Y. Guo, Y. Chen, Y.-K. Wang, M. M. H. H. Li, and M. Gabbouj, "Error resilient coding and error concealment in scalable video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 6, pp. 781–795, Jun. 2009.

[64] S. Shirani, F. Kossentini, and R. Ward, "Reconstruction of motion vector missing macroblocks in H.263 encoded video transmission over lossy networks," in *Proceeding of IEEE International Conference on Image Processing (ICIP)*, Chicago, USA, Oct. 1998, pp. 487–491.

[65] S. Aign and K. Fazel, "Temporal and spatial error concealment techniques for hierarchical MPEG-2 video codec," in *Proceeding of IEEE International Conference on communication*, Seattle, USA, Jun. 1995, pp. 1778–1783.

[66] B. Yan and Gharavi, "A hybrid frame concealment algorithm for H.264/AVC," *IEEE Transactions on Image processing*, vol. 19, pp. 98–107, Jan. 2010.

[67] M. Podolsky, S. McCanne, and M. Vetterli, "Soft ARQ for layered streaming media," *Journal of VLSI Signal Processing Systems*, vol. 27, pp. 81–97, Feb. 2001.

[68] P. A. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," *IEEE Transactions on Multimedia*, vol. 8, pp. 390–404, Apr. 2006.

[69] Z. Miao and A. Ortega, "Expected run-time distortion based scheduling for delivery of scalable media," in *Proceeding of International Packet Video Workshop*, Pittsburgh, USA, Apr. 2002.

[70] *Extended video procedures and control signals for H.300 series terminals.* ITU-T Recommendation H.241, ITU-T, May 2006.

[71] S. Fukunaga, T. Nakai, and H. Inoue, "Error resilient video coding by dynamic replacing of reference pictures," in *Proceeding of Global Telecommunications Conference*, London, United Kingdom, Nov. 1996, pp. 1503–1508.

[72] M. Wada, "Selective recovery of video packet loss using error concealment," *IEEE Journal on Selected Areas in Communications*, vol. 7, pp. 807–814, Jun. 1989.

[73] P.-C. Chang and T.-H. Lee, "Precise and fast error tracking for error-resilient transmission of H.263 video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, pp. 600–607, Jun. 2000.

[74] A. Naghdinezhad and F. Labeau, "Reference frame modification methods in scalable video coding (SVC)," in *Workshop on Multimedia Signal Processing (MMSP)*, Saint Malo, France, Oct. 2010, pp. 200–205.

[75] ——, "An error resilient technique for temporal and spatial scalability," in *Proceeding of The 2011 IEEE International Conference on Multimedia and Expo (ICME 2011)*, Barcelona, Spain, July 2011, pp. 1–6.

[76] ——, "Distortion estimation for reference frame modification methods," in *Proceeding of The 2011 European Signal Processing Conference (EUSIPCO)*, Barcelona, Spain, Aug. 2011, pp. 1–5.

[77] ——, "Reference frame modification techniques for temporal and spatial scalability," *Signal Processing: Image Communication*, vol. 27, no. 10, pp. 1079 – 1095, 2012.

[78] J. Reichel, H. Schwarz, and M. Wien, "Joint scalable video model 11 (JSVM 11)," JVT-X202, Jul. 2007.

[79] H. Schwarz, D. Marpe, and T. Wiegand, "Analysis of hierarchical B pictures and MCTF," in *Proceedings of IEEE International Conference on Multimedia and Expo, ICME*, Toronto, Ontario, Canada, Jul. 2006, pp. 1929–1932.

[80] H. Schwarz, D. Marpe, and T. Wiegand, "Basic concepts for supporting spatial and SNR scalability in the scalable H.264/MPEG4 AVC extension," in *Proceedings of IEEE International Workshop on Systems, Signals and Image Processing*, Greece, Sep. 2005.

[81] M. Wien, H. Schwarz, and T. Oelbaum, "Performance analysis of SVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1194 –1203, Sep. 2007.

[82] Y.-K. Wang, M. Hannuksela, S.Pateux, A. Eleftheriadis, and S. Wenger, "System and transport interface of SVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1149–1163, Sep. 2007.

[83] T. Wiegand, N. Farber, K. Stuhlmuller, and B. Girod, "Error-resilient video transmission using long-term memory motion-compensated prediction," *IEEE Journal on Selected Areas Communication*, vol. 18, pp. 1050–1062, Jun. 2000.

[84] C. Zhu, Y.-K.Wang, and H. Li, "Error resilient video coding using flexible reference frames," in *Proceedings of SPIE Visual Communications and Image Processing, VCIP*, Pittsburg, USA, Jul. 2005, pp. 691–702.

[85] G. J. Sullivan, "Multi-hypothesis motion compensation for low bit-rate video coding," in *Proceeding of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toulouse, France, Apr. 1993, pp. 437–440.

[86] S. Lin and Y. Wang, "Error resilience property of multihypothesis motion compensated prediction," in *Proceeding of IEEE International Conference on Image Processing (ICIP)*, Jun. 2002, pp. 545–548.

[87] W.-Y. Kung, C.-S. Kim, and C.-C. Kuo, "Analysis of multihypothesis motion compensated prediction (mhmcp) for robust visual communication," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, pp. 146–153, Jan. 2006.

[88] R. Satyan, F. Labeau, and K. Rose, "Optimal mode switching for multi-hypothesis motion compensated prediction," in *Workshop on Multimedia Signal Processing (MMSP)*, Saint Malo, France, Oct. 2010, pp. 212–216.

[89] T. Weigand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Error resilient video coding using b pictures in H.264," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 560–576, Jul. 2003.

[90] Z. He and H. Xiong, "Transmission distortion analysis for real-time video encoding and streaming over wireless networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, pp. 1051–1062, Jul. 2006.

[91] 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, pp. 657–673, Jul. 2003.

[92] Z. Wu and J. Boyce, "Adaptive error resilient video coding based on redundant slices of H.264/AVC," in *Proceedings of IEEE International Conference on Multimedia and Expo, ICME*, Beijing, China, Jul. 2007, pp. 2138–2141.

[93] B. Katz, S. Greenberg, N. Yarkoni, N. Blaunstien, and R. Giladi, "New error-resilient scheme based on fmo and dynamic redundant slices allocation for wireless video transmission," *IEEE Transactions on Broadcasting*, vol. 53, pp. 308–319, Mar. 2007.

[94] T. Ogunfunmi and W. Huang, "A flexible macroblock ordering with 3D MBAMAP for H.264/AVC," in *Proceedings of IEEE International Symposium on Circuits and Systems*, Kobe, Japan, May 2005, pp. 3475–3478.

[95] Y.-K. Wang, M. M. Hannuksela, and M. Gabbouj, "Error resilient video coding using unequally protected key pictures," in *Proceedings of International Workshop on Very Low Bitrate Video, VLBV*, Madrid, Spain, Sep. 2003, pp. 290–297.

[96] S. Rane, P. Baccichet, and B. Girod, "Modeling and optimization of a systematic lossy error protection system based on H.264/AVC redundant slices," in *Proceedings of Picture Coding Symposium, PCS*, Beijing, China, Apr. 2006.

[97] Y.-K. Wang and M. Hannuksela, "SVC feedback based coding," Nokia Corporation, Doc. JVT-W052, San Jose, USA, Apr. 2007.

[98] C. He, H. Liu, H. Li, Y.-K. Wang, and M. Hannuksela, "Redundant picture for SVC," USTC and Nokia Corporation, Doc. JVT-W049, San Jose, USA, Apr. 2007.

[99] A. Eleftheriadis, S. Cipolli, and J. Lennox, "Improved error resilience using frame index in NAL header extension for SVC," Layered Media, Inc., Doc. JVT-V088, Marrakech, Morocco, Jan. 2007.

[100] 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, pp. 425–450, Apr. 2006.

[101] S. Wenger, "H.264/AVC over IP," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 645–656, Jul. 2003.

[102] T. Wiegand, G. J. Sullivan, G. Bjntegaard, 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, 2003.

[103] A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video compression," *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 23–50, Nov. 1998.

[104] Y. Shen, P. C. Cosman, and L. Milstein, "Video coding with fixed length packetization for a tandem channel," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, pp. 273–288, Feb. 2006.

[105] B. Girod, "Rate-constrained motion estimation," in *Proceedings of SPIE visual Communications and Image Processing*, vol. 2308, Nov. 1994, pp. 1026–1034.

[106] H. Yang and K. Rose, "Rate-distortion optimized motion estimation for error resilient video coding," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 2, Philadelphia, USA, Mar. 2005, pp. 173–176.

[107] W. N. Lie, Z. W. Gao, W. C. Chen, and P. C. Jui, "Error resilient motion estimation for video coding," in *Advances in Image and Video Technology*, ser. Lecture Notes in Computer Science (LNCS), 2006, vol. 4319, pp. 988–996.

[108] G. Cote, S. Shirani, and F. Kossentini, "Optimal mode selection and synchronization for robust video communications over error-prone networks," *IEEE Journal on Selected Areas Communication*, vol. 18, no. 6, pp. 952–965, Jun. 2000.

[109] "H.264 reference software (ver JM 18.3)," *[Available Online] http://iphome.hhi.de/suehring/tml/.*

[110] H. Yang and K. Rose, "Optimizing motion compensated prediction for error resilient video coding," *IEEE Transactions on Image Processing*, vol. 19, no. 1, pp. 108–118, 2010.

[111] ——, "Advances in recursive per-pixel end-to-end distortion estimation for robust video coding in H.264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, pp. 845–856, Jul. 2007.

[112] M. Fumagalli, M. Tagliasacchi, , and S. Tubaro, "Improved bit allocation in an error resilient scheme based on distributed source coding," in *Proceeding of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Montreal, Canada, May 2004, pp. 61–64.

[113] H. Yang and L. Lu, "A novel source-channel constant distortion model and its application in error resilient frame-level bit allocation," in *Proceeding of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Montreal, Canada, May 2004, pp. 277–280.

[114] Y. Guo, Y.-K. Wang, and H. Li, "Error resilient mode decision in scalable video coding," in *Proceedings of the International Conference on Image Processing (ICIP)*, Atlanta, USA, Oct. 2006, pp. 2225–2228.

[115] D. J. Connor, "Techniques for reducing the visibility of transmission errors in digitally encoded video signals," *IEEE Transactions on Communication*, vol. 21, no. 6, pp. 695–706, 1973.

[116] S. Han and B. Girod, "Robust and efficient scalable video coding with leaky prediction," in *Proceeding of IEEE International Conference on Image Processing (ICIP)*, Rochester, USA, Sep. 2002, pp. 41–44.

[117] W.Peng and Y. Chen, "Error drifting reduction in enhanced fine granularity scalability," in *Proceeding of IEEE International Conference on Image Processing (ICIP)*, Rochester, USA, Sep. 2002, pp. 61–64.

[118] R. Satyan, S. Nyamweno, and F. Labeau, "Novel prediction schemes for error resilient video coding," *Signal Processing: Image Communication*, vol. 25, no. 9, pp. 648 – 659, 2010.

[119] ——, "Error resilience using leaky source channel prediction scheme," in *Proceeding of IEEE International Conference of Computer Communications and Networks*, San Francisco, USA, Aug. 2009, pp. 1–5.

[120] Y. Wang, S. Wenger, and M. Hannuksela, "Common conditions for SVC error resilience testing," ISO/IEC JTC 1/SC29/WG 11, JVT-P206d1, Jul. 2005.

[121] "H.264/SVC reference software (ver JSVM 9.15) and manual," *[Available Online] CVS sever at garcon.ient.rwth-aachen.de.*

[122] H. L. Y. Guo and Y. Wang, "SVC/AVC loss simulator donation," ISO/IEC JTC1/SV29/WG11 and ITU-T SG16 Q.6, Document JVTP069, 2005.

[123] S. Wenger, "Error patterns for internet experiments," VCEG Q15-I-16r1, 2002.

[124] M. M. Hannuksela, Y.-K. Wang, and M. Gabbouj, "Sub-picture: ROI coding and unequal error protection," in *Proceeding of IEEE International Conference on Image Processing (ICIP)*, Rochester, USA, Sep. 2002, pp. 537–540.

[125] A. Naghdinezhad, A. Farmahini-Farahani, M. Hashemi, and O. Fatemi, "An adaptive unequal error protection method for error resilient scalable video coding using particle swarm," in *Proceeding of the IEEE International Conference on Signal Processing and Communications (ICSPC)*, Dubai, UAE, Nov. 2007, pp. 396 –399.

[126] Y. Wang, T. Fang, L. P. Chau, and K. H. Yap, "Two-dimensional channel coding scheme for MCTF-based scalable video coding," *IEEE Transactions on Multimedia*, vol. 9, no. 1, pp. 37–45, Jan. 2007.

[127] B. Zhang, M. Wien, and J.-R. Ohm, "A novel framework for robust video streaming based on H.264/AVC MGS coding and unequal error protection," in *Proceeing of the IEEE International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, Kanazawa, Japan, Dec. 2009, pp. 107–110.

[128] A. Naghdinezhad, M. Hashemi, and O. Fatemi, "A novel adaptive unequal error protection method for scalable video over wireless networks," in *Proceeding of the IEEE International Symposium on Consumer Electronics (ISCE)*, Dallas, USA, Jun. 2007, pp. 1 –6.

[129] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceeding of the IEEE International Conference on Neural Networks*, vol. 4, Perth, Australia, Nov. 1995, pp. 1942–1948.

[130] F. Zhai, Y. Eisenberg, C. E. Luna, T. N. Pappas, R. Berry, and A. K. Katsaggelos, "Packetization schemes for forward error correction in internet video streaming," in *Proceeding of the 41st Allerton Conference Communication, Control and Computing*, Monticello, USA, Oct. 2003.

[131] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*. Prentice Hall, 1983.

[132] A. Albanese, J. Blomer, J. Edmonds, M. Luby, and M. Sudan, "Priority encoding transmission," *IEEE Transactions on Information Theory*, vol. 42, pp. 1737–1744, 1996.

[133] R. Gallager, *Low Density Parity-Check Codes*. MIT Press, Cambridge,MA, 1963.

[134] D. E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*. Addison Wesley, Dec. 1983.

[135] G. Hasslinger and O. Hohlfeld, "The Gilbert-Elliott model for packet loss in real time services on the internet," in *Conference on Measuring, Modelling and Evaluation of Computer and Communication Systems*, Dortmund, Germany, Apr. 2008, pp. 1 –15.