Multi-State Video Coding over Error Prone Channels

vorgelegt von
Diplom-Ingenieurin
Sila Ekmekci
aus Frankfurt am Main

Von der Fakultät IV - Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
-Dr.-Ing.-

genehmigte Dissertation

Promotionsausschuß:

Vorsitzender: Prof. Dr.-Ing. O. Hellwich
Berichter: Prof. Dr.-Ing. T. Sikora
Berichter: Prof. Dr.-Ing. P. Noll

Tag der wissenschaftlichen Aussprache: 22. Dezember 2004

Berlin 2004
D 83

Preface

The work investigates Multi-State Video Coding (MSVC) which is a Multiple Description Scheme. MSVC is interesting because of its low complexity and its low delay property which makes it attractive for streaming applications. The performance of MSVC is explored in terms of the average PSNR of the reconstructed video sequence. We compare MSVC to Single-State Video Coding (SSVC) and Temporal Layered Coding (TLC) under different channel conditions and reception scenarios. Moreover we investigate the trade-off between rate allocated to quantization accuracy and to intra-coding in terms of its effect on the average PSNR over error-prone channels. Besides balanced MSVC, unbalanced MSVC by adaptation of quantization is analyzed. It was shown under which conditions it is practical to switch to unbalanced operation from the balanced one. Moreover, an improved version of the original MSVC approach is developed where state recovery is not only used for error concealment in case of losses but also then whenever it enables a larger frame PSNR. It was shown that the improved version yields better results for all channel conditions and rate allocations. In the second part of the work, a recursive decoder distortion estimation model based on the AR(1) modeling of the video source is presented and used to support the simulation based results from the theoretical point of view. According to the model, each frame block belongs to a different AR(1) source with all of its corresponding blocks along the video sequence. The distortion of each frame block is calculated dependent on the distortions of its corresponding blocks on the past frames. In the next step, the estimation method is extended from Single-State-Video Coding to Temporal Layered Coding and Multi-State Video Coding. The advantages of the model is its relative low complexity and high estimation accuracy. These properties makes it especially applicable for streaming applications where decoder distortion should be estimated at the server before choosing among the possible streaming options.

Acknowledgments

I would like to thank my advisor Prof. Dr. Thomas Sikora for his encouragement and support in forming this thesis, for fruitful discussions and valuable advices. I thank my second advisor Prof. Dr. Peter Noll for waking interest in me to work in the fields of communications.

I am especially grateful for my research visits at the Broadband Telecommunications Center of Georgia Tech Wireless Institute and Information Systems Laboratory of Stanford University which contributed a lot to my technical education as well as to my personal formation. I owe many thanks to Prof. Dr. Nikil Jayant and Prof. Dr. Bernd Girod for inviting me to their groups, for valuable discussions and for their hospitality. I am grateful to members of Broadband Telecommunications Center of Georgia Tech and of the Information Systems Laboratory of Stanford University for the stimulating and friendly atmosphere I enjoyed much and made my stays at both universities unforgettable.

I also thank my friends and colleagues at Telecommucations Institute of Technical University Berlin for their collaboration in the last years, for their support and help in solving problems. Especially the administrative support of Birgit Boldin is unaffordable.

My special thank goes to my family for giving me invaluable emotional support.

Zusammenfassung

Anwendungen von Verfahren der Multiple Description Coding (MDC) zur Codierung von Signalen haben in den letzten dreissig Jahren grosses Interesse erweckt. In den siebziger Jahren beschäftigten sich die Forscher von Bell Labs mit den theoretischen Aspekten des Verfahrens. Das Grundprinzip ist die Erzeugung und Übertragung von mehreren Beschreibungen für die gleiche Quelle. Wenn nur eine Beschreibung empfangen wird, rekonstruiert der Empfänger das Signal mit einer akzeptablen Qualität. Die Signalqualität wächst mit der Anzahl der empfangenen Beschreibungen und ist maximal, wenn alle Beschreibungen erhalten werden. Der Unterschied zum Layered Coding liegt darin, dass jede empfangene Beschreibung ohne Ausnahme eine Verbesserung der Rekonstruktionsqualität ermöglicht. Im Layered Coding ist die Dekodierung der Erweiterungsbeschreibung abhängig von der Dekodierung der Basisbeschreibung, so dass, wenn die Basisbeschreibung verloren geht, die Erweiterungsbeschreibung nicht mehr dekodiert werden kann und nichts zur Rekonstruktionsqualität beitragen kann. Die Anwendung von Multiple Description Coding auf Videosignale ist wegen kurzen Verzögerungszeiten und niedrigen Kodierungs- und Dekodierungskomplexität besonders erfolgversprechend.

Multi-State Video Codierung (MSVC) ist eine besondere Form von MDC, wobei die Kodierungseffizienz mit der Fehlerresistenz abgewägt wird. Die Videosequenz wird bei zweifacher Beschreibung in zwei Teilsequenzen aufgeteilt, wobei eine aus den gerade Bildern und die andere aus den ungerade Bildern besteht. Die Bewegungsschätzung zur Kompression erfolgt zwischen benachbarten Bildern in jeder Teilsequenz. Die dadurch enstandenen Bitströme werden in Pakete aufteilt und über unterschiedliche Kanäle zum Empfänger übertragen, wodurch Pakete unterschiedlicher Teilsequenzen unterschiedlich fehlerbehaftet sind. Der vergrösserte zeitliche

Abstand zwischen dem Referenz- und dem aktuellem Bild verkleinert den Prädiktionsgewinn. Im Gegensatz dazu wird die Fehlerresistenz des gesamten Systems durch die getrennten Prädiktionsschleifen und getrennten Zustände der beiden Teilsequenzen erhöht. Wenn ein Bild in einer Teilsequenz wegen Paketverlusten nicht empfangen wird, kann die Rekonstruktion durch eine Interpolation der vergangenen und zukünftigen Bilder der anderen Teilsequenz erfolgen. Diese Eigenschaft wird als State Recovery bezeichnet, wodurch die Fehlerresistenz des Systems erhöht wird. Der Endschritt ist das Zusammenfügen der beiden Teilsequenzen, wobei die durch Paketverluste verlorengehenden Bilder durch State Recovery rekonstruiert werden, so dass die Dekodierung der nachfolgenden Bilder ermöglicht wird.

Die vorgelegte Arbeit untersucht die MSVC Technik im Hinblick auf das mittlere PSNR (peak signal to noise ratio) bei verschiedenen Paketverlustwahrscheinlichkeiten und Rekonstruktionsmöglichkeiten. Im ersten Teil der Arbeit besteht das Ziel darin, für eine gegebene Gesamtbitrate die optimale Ratenaufteilung zwischen zwei Beschreibungen zu finden, wobei das über alle Bilder gemittelte PSNR maximiert wird. Mit anderen Worten, ausgehend von den gleichratigen Beschreibungen (gleiche Bitrate) werden ungleichratige Beschreibungen untersucht. Um die Bitraten der Beschreibungen zu variieren und gleichzeitig die Gesamtbitrate konstant zu halten, werden die Quantisierungsgenauigkeiten der beiden Teilsequenzen entsprechend eingestellt, was eine der Möglichkeiten ist, um ungleichratige Beschreibungen zu generieren. Die Untersuchungsergebnisse haben gezeigt, dass, man wenn die für die Übertragung benutzten Kanäle mit unterschiedlicher Fehlerrate behaftet sind – vorausgesetzt, dass man die Raten durch "probing" vordetektieren kann – durch ungleichratige Beschreibungen Gewinne im mittleren PSNR erzielen kann, wobei der Gewinn um so grösser ist, je bewegungsreicher die Videosequenz und je unterschiedlicher die Fehlerraten der beiden Kanäle sind. Im nächsten Schritt wird die Ratenaufteilung zwischen Quantisierungsgenauigkeit und Intra-Kodierung untersucht. Die Ergebnisse haben gezeigt, dass Intra-Kodierung wichtiger als die Quantisierungsgenauigkeit ist, wenn die Kanalfehlerrate gross und wenn die Videosequenz bewegungsintensiv ist. Im Gegensatz dazu ist die Intra-Kodierung für bewegungsarme Sequenzen nicht mehr effizient, weil intra-codierte Blöcke die Bitrate erhöhen und wegen der langsamen Bewegung für die Fehlerresistenz keine grosse Rolle spielen. Eine andere Beobachtung im Hinblick auf die Ergebnisse ist, dass im Falle der ungleichen Bitraten, wenn die

Fehlerrate des schlechteren Kanals erhöht wird, das mittlere PSNR Wert grösser wird. Der Grund ist, dass die Interpolation der mit hoher Rate quantisierten Bilder bessere Ergebnisse ergibt als die Verwendung der empfangenen aber mit niedriger Bitrate kodierten Bilder. Die Anzahl der benötigten Interpolationen steigt mit der wachsenden Fehlerrate. Ausgehend von dieser Beobachtung wurde eine weitere Methode für MSVC entwickelt, wobei das Bild des niedrigratigen Stroms durch das interpolierte Bild ersetzt wird, auch wenn das Bild empfangen wurde, aber die Rekonstruktion durch Interpolation ein höheres PSNR ergibt. Um zu garantieren, dass die Fehlerfortpflanzung die Rekonstrunktion des nächsten Bildes in der Teilsequenz nicht verschlechtert, wird der Effekt der Fortpflanzung auch in die Entscheidung der Rekonstruktionsmethode herangezogen. Der Vergleich zwischen der ursprünglichen MSVC Technik und der erweiterten Technik zeigt, dass das PSNR fast in allen Rateaufteilungen und Kanalbedindungen durch die neue Technik erhöht wird.

Im nächsten Schritt wird MSVC mit Verfahren der Single-State Video Coding (SSVC) and Temporal Layered Video Coding (TLC) verglichen. Dabei wurden zwei Szenarien betrachtet: In dem ersten Szenario wird angenommen, dass in SSVC und TLC die Bewegungsvektoren immer erhalten bleiben (z.B. durch gezielte Kanalkodierung), auch wenn die dazugehörige Bildinformation verloren geht. Im zweiten Fall wird keine Bewegungsinformation für verlorengegangene Bilder vorausgesetzt. Im ersten Fall wird eine bewegungsbasierte Fehlerverschleierung verwendet, während im zweiten Fall das verlorengegangene Bild durch Wiederholung des zurückliegenden Bildes ersetzt wird. Die Ergebnisse haben gezeigt, dass im ersten Fall sowohl TLC als auch SSVC MSVC übertreffen, während im zweiten Fall MSVC trotz kleinerem Prädiktionsgewinn vorteilhaft ist, weil die Rekonstruktion durch Interpolation bessere Ergebnisse liefert als mit Bildwiederholung. Der Vorteil vom MSVC wird mit wachsender Fehlerrate deutlicher.

Im zweiten Teil der Arbeit wird untersucht, wie die Rekonstruktionsfehler theoretisch erfasst werden koennen. Die Fehler setzen sich aus den Quantisierungsfehlern, aber auch aus Verschleierungsfehlern und deren Fortpflanzungen über die Zeit zusammen. Ein Schätzmodell wurde entwickelt, mit dessen Hilfe der Gesamtfehler jedes Bildes mit einer hohen Genauigkeit und geringer Komplexität geschätzt werden kann, wenn die Fehlerrate der Kanäle und die Quantisierungsfehler vorgegeben werden. Die Methode wurde zuerst für Fehlerschätzung bei einer SSVC-Übertragung

entwickelt und danach auf TLC und MSVC-Übertragung erweitert. Im Vergleich zu den Fehlerschätzungstechniken, die in der Literatur bekannt sind, ist die entwickelte Methode besonders interessant, weil die Algorithmen, die genaue Ergebnisse liefen, hohe Komplexität haben und nur unter bestimmten Voraussetzungen angewendet werden können. Die Techniken mit niedriger Komplexität, andererseits, gelten nur im Falle von einzelnen Paketverlusten, die einen bestimmten Abstand voneinander haben. Für Folgefehler sind solche Techniken nicht geeignet, besonders weil die Fortpflanzung durch Folge-Paketverluste mit einfachen Modellen schlecht erfassbar und das Prinzip der additiven Überlagerungen nicht direkt anwendbar ist. Unsere Methode beruht auf eine AR(1)-Modellierung jeder Sequenz von korrespondierenden Blöcken auf nachfolgenden Bildern entlang der Zeitachse. Basierend auf den Empfangswahrscheinlichkeiten des Referenzbildes, des aktuellen Bildes und der benachbarten Bilder auf der anderen Teilsequenz, werden verschiedene Fälle untersucht, und für jeden Fall ein Fehlerterm rekursiv ausgerechnet. Die mittlere Verzerrung wird dann über alle möglichen Fälle gemittelt, wobei die Schätzung blockweise erfolgt. Im nächsten Schritt wurde das Verfahren für mögliche Block- und Bild-Intra-Updates erweitert. Die Technik ist besonders nützlich, da beim adaptiven Video-Streaming z.B. an den Proxy Servern zwischen verschiedenen Übertragungsmodi umgeschaltet werden kann. Um den richtigen Übertragungsmodus auszuwählen, muss aber der Rekonstruktionsfehler für jeden Modus in Abhängigkeit von dem Kanalzustand berechnet werden. Dafür ist wiederum eine genaue und trotzdem nicht komplexe Schätzungstechnik erforderlich.

Zusammenfassend untersucht die Arbeit praktische und theoretische Probleme der Multi-State Video Coding. Zusätzlich wurde eine Erweiterung der Methode vorgeschlagen, die für alle Paketverlustraten und alle Ratenaufteilungen Verbesserungen erzielt. Multi-State Video Coding wurde von gleichratigen Codes auf ungleichratige Codes erweitert, und es wurde gezeigt, unter welchen Umständen unausgeglichene Codes Gewinne bringen. Multi-State Video Coding wurde durch Simulationen mit Single-State Video Coding und Temporal Layered Coding verglichen, und es wurde herausgefunden, wann Multi-State Video Coding die beiden anderen Verfahren übertrift. Um theoretische Aussagen über das Verhalten von verschiedenen Übertragungsmöglichkeiten in unterschiedlichen Kanalsituationen machen zu können, wird abschliessend eine neue Fehlerschätzungstechnik entwickelt, die sich durch ihre

niedrige Komplexität und hohe Schätzgenauigkeit auszeichnet und sich dadurch von bisher bekannten Methoden in der Literatur unterscheidet.

Contents

Chapter 1

Introduction

Application of Multiple Description (MD) Techniques for coding of signals attracted big interest in the recent thirty years. Video Coding using Multiple Description is interesting because of the low coding delay which plays an important role, especially in streaming applications.

In this thesis, Multi-State Video Coding (MSVC), a special scheme for MD coding is explored in terms of its trade-off between prediction gain and error resilience. MSVC basically divides the video sequence into two subsequences consisting of the odd numbered and even numbered frames each. The motion compensated frame prediction occurs between the subsequent odd and even frames in each subsequence. The resulting bitstreams are divided into packets and the sequence of packets for each subsequence is transmitted each over a different channel to the receiver. Each channel may incur a different loss pattern corresponding to a different loss probability. According to this, the prediction gain is decreased due to the increased distance between adjacent frames in each subsequence. On the other hand, due to the independent loops and states, the prediction dependency between the two subsequences is broken with the consequence if one of the frames in one of the subsequences is lost, the previous and next frames from the other subsequence can be used to recover from losses in the current subsequence. This property of MSVC is referred to as state recovery. After receiving the two streams and applying state recovery in case of lost frames, the reconstructed subsequences are interleaved to generate the final reconstructed sequence. In evaluating the system performance, we are especially interested in the average peak signal to noise ratio, $PSNR_{avg}$ of the reconstructed

sequence at differing loss probabilities of the transmission channels used.

In chapter 2, the general MD techniques besides MSVC are presented. Among the techniques outlined, MSVC is advantageous due to its low complexity of implementation and reconstruction and also its low delay property. The original MSVC Approach presented in [4] is called Approach 1 and is discussed in section 3.4. First, Encoder and Decoder Portions of the System are presented in sections 3.1 and 3.2 and then State Recovery in case of losses is discussed in section 3.3. $PSNR_{avg}$ is investigated in case of lossy behavior of only one channel in section 3.4.1 and of both channels in section 3.4.2. The effects of intra updates (updates of Group of Blocks (GOB's) as well as frames) on the MSVC scheme are explored in section 3.4.3 where the total bitrate used to encode the two subsequences R_T is kept constant. To adjust the trade-off in bitrate allocation, intra-update periods and encoder quantization stepsizes are varied. It is shown that intra-updates are more important than small quantization errors for high motion sequences at increasing loss rates. For low motion sequences, on the other hand, intra-updates are not efficient at all. Error concealment by simply techniques yield good results because of the low motion.

In section 3.4.4, the interest is focused on unbalanced MSVC. The available bitrate is divided unequally between the encodings of the two subsequences. There are basically three methods for unbalanced MSVC: 1-based on frame rate adaptation, 2-based on quantization adaptation and 3-based on the spatial resolution. We concentrate in this work on unbalancing based on quantization adaptation. Specifically, while the encoder quantization step size of one of the subsequences is decreased the quantization step size of the other one is increased so that the total bitrate R_T is kept constant. We explore here in which scenarios unbalanced quantized MSVC makes sense in terms of maximization of $PSNR_{avg}$. It is shown that unbalanced coding and transmission is useful and yields better performance when the loss probabilities of channels in use are different.

Section 3.5 introduces Approach 2 for MSVC. The idea is inspired from the state recovery aspect of MSVC: Even if a frame is received, if the interpolated frame at the same temporal location using the previous and next frames from the other stream yields a higher frame PSNR than reconstructing the frame from the received packet, state recovery is to be employed although the frame is not lost. In section 3.6, Approach 1 is compared to Approach 2 at different channel conditions. It is shown

that Approach 2 yields better results than Approach 1 for all channel conditions and rate allocations.

In chapter 4, we compare MSVC to Single-State Video Coding (SSVC) as well as to Temporal Layered Coding (TLC). We use two channels for MSVC and TLC and just one channel for transmission of SSVC packets. We differentiate between two cases: First we assume that motion vectors are always received at the decoder, e.g. by using channels. In the second case, we consider that the motion vectors are also lost with the corresponding frame information. We show that MSVC outperforms both SSVC and TLC if the motion vectors are not received and as the channel losses are increasing. SSVC and TLC, on the other hand perform better if the motion information is received and the error concealment is eased.

In the sequel of the work, we verify the simulation based results via a model of the representation and reconstruction of the video signal. Chapter 5 is dedicated to decoder distortion estimation for single-state video transmission. The algorithm divides each frame into 4x4 blocks. Each set of blocks with their corresponding blocks along the sequence is assumed to correspond to a different AR(1) source each with a different correlation coefficient. The distortion of each block in each frame is estimated separately where the estimation is performed recursively depending on the distortions of the corresponding blocks in the previous frames. In sections 5.2.2 and 5.2.3, the estimation technique is extended to cover also the intra-updates of GOB's and frames respectively. The verification of the model via comparison to simulation results is given in sections 5.2.2 and 5.2.3.

The decoder distortion estimation method is extended to Layered and Multi-State Video Coding in chapters 6 and 7. We discuss the model and distortion estimation algorithm and present the comparison of the simulation and model results. The model is especially attractive for streaming applications because of its low complexity and high estimation accuracy. The algorithms known from the literatur so far, are divided into two: Those yielding high accuracy at the cost of high calculation complexity and those having low complexity but also low accuracy and restricted application area. The technique presented here gives high accuracy results for an acceptable complexity and moreover it is applicaple for any sort of loss patterns and coding options.

Chapter 8 deals with the methods of achieving path diversity so that two or more streams can be transmitted over different channels with possibly different transmission

characteristics. Although this topic is not the main focus of the work, the chapter gives an outlook about the current possibilities. Chapter 9 concludes the work.

Chapter 2

Multiple Description Coding

Current systems typically generate content with a progressive coder (e.g. as in [120]) and deliver it with TCP which controls retransmission of the lost packets. Both of these techniques together can produce large delays when packets are lost. Progressive transmission allows the quality to improve steadily as the number of consecutive packets are received. Embedded Coding and JPEG2000 standard using the concept of scalability are presented in [141], [129] and [151]. When the packets are sent and received in order without losses, progressive transmission works well. But the reconstruction stops until the particular packet in the sequence is received if there are losses. The delay in receiving a retransmitted packet may be much longer than the interarrival times between received packets. TCP-based content delivery suffers from this. The weakness of the progressive transmission system is that the source coding in the conventional system puts too much faith in the transmission, i.e. the received packets are only useful if all earlier packets have been received. A variety of techniques have been proposed to enhance the robustness of the video communication system against packet losses. Some of them are presented in [121], [6], [103], [137], [157], [136], [156], [73], [66] and [159].

The idea behind Multiple description (MD) coding is: If losses are inevitable, representations which make all received packets useful can be of great benefit. MD coding trades off robustness against the loss of descriptions and compression efficiency, therefore it is only to be applied if the trade-off pays off. MD coding was invented at Bell Laboratories in connection with communicating speech over the telephone network. Today it is applicable to image, video or audio transmission. A good

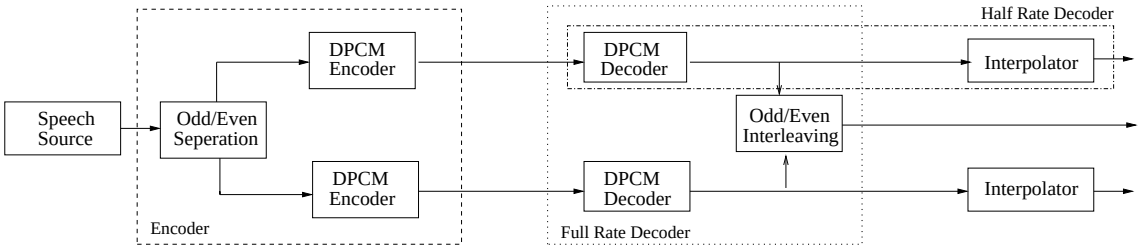


Figure 2.1: Speech coding for channel splitting as proposed by Jayant

summary of the development of the MD approach and the major MD coding methods are given in [53].

The outages of transmission links are inevitable. They arise from device failures as well as from routine maintenance and upgrades. To achieve high reliability, mechanisms are required to handle outages. The initial idea was to split the information from a single call and send it on two separate links or paths. If everything works, both halves of the transmitted data could be combined for the usual voice quality. If one of the links undergoes an outage, the other would enable the communication at reduced quality. The first references from the 1970's are [46], [100], [99] and [16]. The ideas presented in [100] and [99] are related to Gray coding [122].

MD idea was first investigated for analog speech transmission. Speech sampled at the Nyquist rate can be subsampled by two without too much aliasing because of its decaying frequency spectrum. According to this, odd-numbered samples are sent on one channel and even-numbered samples on the other one. In 70's channel splitting become popular for speech coders and information theorists. Jayant's speech coding method for channel splitting is proposed in [78] and a similar system for packet-switched telephony in [80]. Figure 2.1 shows the channel splitting method for speech coding by Jayant.

A speech signal is bandlimited to 3.2 kHz and sampled at 8 kHz in telephony. Jayant in his system uses the initial subsampling of 12 kHz to reduce aliasing after subsampling by two for channel splitting. Differential Pulse Code Modulation (DPCM) is used to compress the odd and even numbered samples separately which are sent on two separate channels. A DPCM decoder for each channel is required and the samples are interleaved to produce a signal with 12kHz sampling summed with some amount of DPCM quantization noise. Adaptive Linear Interpolation is used to

decode from a single channel.

The idea of scalar quantization for channel splitting was first investigated by Reudink [117] but reinvented and analyzed later by Vaishampayan [147]. Reudink was the first to propose channel splitting techniques that do not increase the total rate so much and do not entirely rely on preexisting redundancy in the source sequence. Gersho, on the other hand proposed the use of modulo-PCM encoding [40] for channel splitting [46]. Goodmans suggestion was one channel to carry the most significant bits of the even-numbered samples and the least significant bits of the odd-numbered samples. This was later studied in [112].

Parallel to studies in the area of speech coding, channel splitting was also studied in context of information theory. The question asked by the information theoretists was: If an information source is described by two separate descriptions, what are the concurrent limitations on qualities of these descriptions taken separately and jointly? The MD situation is described as follows: An encoder is given a sequence of source symbols $\{X_k\}_{k=1}^{N}$ to communicate to three receivers over two noiseless channels. The central decoder receives information from both channels whereas both side decoders receive information over their respective channels. R_i, i =1,2, is the transmission rate over Channel i in bits per source sample. $\{\hat{X}_k^{(i)}\}_{k=1}^{N}$ is the reconstruction sequence produced by Decoder i associated with the distortions D_i, i =0,1,2. The difficulty of MD coding is in its conflicting requirements. Two good descriptions at rates R_1 and R_2 together yield not necessarily a good description for the total rate $R_1 + R_2$. Likewise, it is not easy to generate two good representations from a good description (in terms of good compression vs. quality tradeoff) at rate $R_1 + R_2$. The fundamental tradeoff of MD coding is making descriptions individually good yet not too similar. The central problem in MD modeling is to determine for a given source and distortion measure, the set of achievable values for the quintuple $(R_1, R_2, D_0, D_1, D_2)$. MD with two descriptions is shown schematically in Figure 2.2.

The early theoretical MD work was focused on coding memoryless binary sources with no more total bits than necessary to generate one single description for the same source. The interest in MD literature as in [3], [11], [161], [163], [164] and [171] was on the memoryless binary symmetric source (BCC) with Hamming distortion.

Successive Refinement (SR) Coding [38] is a special case of MD coding. In SR, channel 1 is received by all decoders and channel 2 is received by only one decoder.

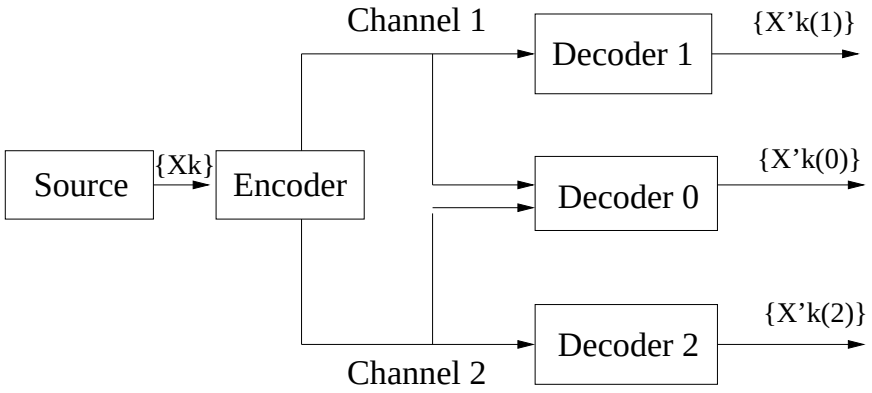


Figure 2.2: MD source coding with two channels and three receivers

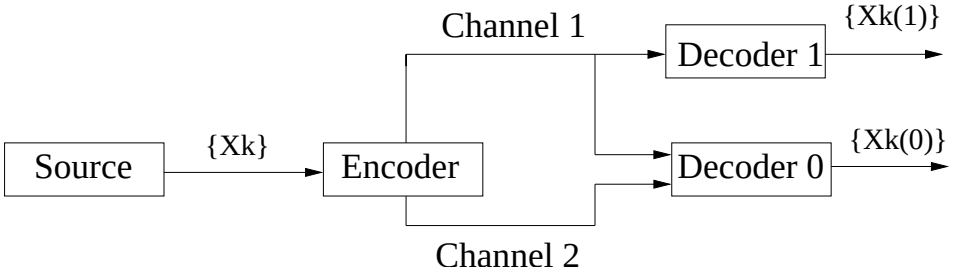


Figure 2.3: The successive refinement problem

Channel 2 is used then as refinement and is useless without Channel 1 [39], [85], [86], [87], [118]. If different users are associated with different decoders, SR coding applies to layered broadcasting [113], [96]. The idea behind SR coding is that the most important data is sent on Channel 1 and additional data is sent on Channel 2 to improve the reconstruction quality: i.e. the most significant bits can be sent on Channel 1 and least significant ones on Channel 2 or similarly low pass version on Channel 1 and high pass version on Channel 2. Figure 2.3 depicts the successive refinement problem schematically.

Another network problem concerning MD problem was introduced by Gray and Wyner: Instead of having a single source sequence to communicate over two channels to three receivers, they have a sequence of pairs of random variables $\{X_k, Y_k\}_{k=1}^{N}$ to communicate to two receivers over three channels. Whereas Receiver 1 is only interested in $\{X_k\}$ receiver 2 is interested in the other sequence $\{Y_k\}$ [65]. MD coding is expected to be useful in the following conditions: 1- One or more users sometimes fail to receive one or more descriptions 2- Various quality levels (distortions) are

acceptable and distinguishable.

MD rate distortion region is investigated in [45], [150], [171], [105], [167], [162], [93]. The capacity definitions for channels in context of MD is discussed in [33]. The distortion rate functions of Gaussian sources can be computed easily for any rate [10], [24], [63]. Coding theorems for discrete sources are given in [127].

MD codes can be generated based on prediction, quantization, decorrelating transforms and entropy coding. One method is to partition the source data into several sets and then compress them independently to produce descriptions. To decode from any proper subset of descriptions, interpolation is used, e.g. like in channel splitting. The effectiveness of the technique is dependent on the redundancy of the source. There are different other methods to generate MD codes independent of the redundancy. Modern techniques use prediction and decorrelating transforms afterwards, i.e. MD codes can be designed using these basic blocks.

The simplest method is to repeat the same information twice, i.e. the two descriptions are the same. The problem is that nothing is gained if both descriptions are received. To reduce the necessary bitrate we could only repeat one part of the data, i.e. the more important part. The scheme may look like that: 1-first encode to a rate $(2-\zeta)R$ with a progressive source coder with $\zeta \in [0,1]$. The first ζR bits are the most important and thus are repeated in both descriptions. The remaining $2(1-\zeta)R$ bits can be split between the descriptions. According to this, some bits are protected with a rate 1/2 channel code and the other bits are unprotected. This scheme is called Unequal Error Protection, UEP, which generalizes easily to more than two descriptions. To produce L descriptions, use channel codes with rates 1/L, 2/L,...1. UEP design includes the decision about how much of the data to code at each channel code rate [101], [110]. Quantizers can also be used to produce two complementary descriptions of the same scalar quantity each with a 2-Bit quantizer, the central decoder corresponding to $(B+1)$ bit quantizer. This means that $2B$ bits are used to generate a $(B+1)$ bit resolution. Some of works done in this area by Reudink [117] and Vaishampayan are given in [142], [143] and [144]. The advantage of MD scalar quantization is its flexibility in choosing the relative importance of the central distortion and each side distortion. In the balanced case we have $R_1 = R_2$ and $D_1 = D_2$. Reference [9] analyzes the application of MD scalar quantization to transform coefficients.

MD vector quantization is an extension of MD scalar quantization, but the problem is the difficulty in index assignment because the code vectors cannot be naturally ordered. MD lattice vector quantization presented by Servetto in [126] and [148] avoid these problems. The difficulty of optimizing vector quantizers is discussed in [32], [47] and [64]. Lattice structure reduces the encoding complexity [84], [56], [55] and the index assignment problem is simplified as given in [22]. Diggavi extended MD Lattice Vector Quantization for unbalanced descriptions by using two different sublattices [28]. References [41], [43] and [77] describe other MD quantization schemes. MD coding based on transforms inserts statistical dependencies between transform coefficients. This way, the lost coefficients can be estimated from the received ones [154]. Let X_1 and X_2 be independent, zero-mean Gaussian random variables with variances σ_1^2 and σ_2^2 and $\sigma_1^2 \neq \sigma_2^2$. The reference [154] uses the descriptions: $Y_1 = 2^{-1/2}(X_1 + X_2)$ and $Y_2 = 2^{-1/2}(X_1 - X_2)$. According to this, (X_1, X_2) can be recovered from both descriptions. Moreover they are correlated with correlation coefficient: $(\sigma_1^2 + \sigma_2^2)^{-1}(\sigma_1^2 \sigma_2^2)$. This idea is extended to more general description generations and longer vectors in [56] and [57]. In [103] and [155], transform coefficients are first quantized and then transformed yielding better performance than producing correlated transform coefficients and quantizing them. Linear transforms are used in [56] and [57]. MD coding with frames was proposed by Goyal in [60]. The idea is similar to block channel codes. The source vector x is left multiplied by a rectangular matrix F to produce M transform coefficients where x having the dimension N. From M transform coefficients L descriptions are generated after the quantization. The multiplication by F is called frame expansion and the representation is called a quantized frame expansion (QFE) [26], [31]. The estimation of the source vector x from the quantized expansion coefficients $y = Q(Fx)$ can be seen as a least-squares problem: $\hat{x} = argmin_x \|y - Fx\|^2$ [107]. The resulting distortion is proportional to N/M. The accuracy of the estimate is increased by more complicated reconstruction methods [19], [62] and [114]. The frame operator F has the same function as the block channel code. QFE quantizes dependent quantities, so that linear dependence is broken. According to this, each transform coefficient gives some independent information. Extensions of the method are given in [61], [59], [97], [8], [29], [30] and [166]. Discrete transforms can also be used in MD coding which can be obtained in several ways [71], [168] and [15] and can also be used for single description transform coding

[52], [54].

MD coding is especially appropriate for packet networks where packets are directly connected to descriptions. There are different reasons for packet losses in data networks [152]. The packet loss probability can vary widely with time of day, day of the week and connection routing. Another problem cause is that Internet is becoming more heterogeneous as backbone capacities increase and more wireless devices are connected. Packets may be dropped if the capacity of the bottleneck link is exceeded. In networks where selective treatment to some packets is possible, a progressive source code may be feasible. But in networks such as Internet where such services are not available, packets are dropped at random where the use of MD codes pays off. Retransmissions are used in data networks frequently to cope with losses [48]. The receiver has to send positive and/or negative acknowledgements to sender to initiate retransmissions. This method is only useful if the packet losses are rare and thus the additional network overload through retransmission is not that big. Otherwise the retransmission can increase the delay in the network increasing the packet dropping rate contributing back to the main problem. If feedback is not possible retransmission cannot be employed. There are different reasons for not having feedback: It could be expensive, feedback may be causing feedback implosion, e,g. in broadcasting. But the primary problem of feedback is delay. The delay may extend from one round-trip time to arbitrarily large. Delay is a very important parameter if interactivity is required. Size of buffer required for streaming audio or video applications is determined by the transport delay variation.

Another advantage of MD codes over channel codes is avoiding long block sizes creating difficulties associated with delay. It is difficult to attain good performance with FEC. So that FEC becomes effective, channel code output symbols should be placed in different packets. The length of an FEC code is limited to the number of packets used in the communication. As an example, an Internet protocol, version 6 node [27] is required to handle 576 byte packets without fragmentation. Larger packets are recommended to be accommodated. According to that, a typical image can be communicated in about ten packets [67]. Ten is on the other hand too low as the number of output symbols of a channel code. Path diversity where descriptions are sent over different paths to the receiver increase the applicability of MD coding. Use of MD coding is also interesting against high packet loss as in wireless systems.

MD coding requires some diversity but would tend to work with lower diversity than methods based purely on channel coding [165]. Furthermore, the dependencies in descriptions could be used to improve demodulation and decoding performance [130], [81]. The importance of latency requirements is given [128].

To sum up, MD coding does not require the transport mechanism to be flawless. It shifts the task to resist against channel losses to source coding part of the system. In the ideal case, a MD code should make all of the received data useful and the loss of some of the transmitted data not catastrophic.

In literature, MD coding is applied to code audio, images as well as video data. Application on speech coding was started with channel splitting [78]. Further techniques followed such as prediction [76], [145], perceptual models [88], repetition with optimized bit allocations [83] and correlating transforms [7]. The application of MD for radio broadcast is given in [79] and [106]. MD in context of distributed storage is discussed in [25].

Channel splitting was also applied to images in [134]. Another method was correlating transforms in context of image coding [154] followed by other transform based techniques [155], [20] and [82]. Progressive image coders combined with UEP are investigated in [98], [102], [110], [149] and [119]. MD scalar quantizers and quantized frames are applied to images in [125] and [19], [58] respectively.

MD coding is also applied to video. Especially for video streaming, MD codes can be useful because of the low delay property. Among the several papers on MD video coding, [91] uses MD protection for the most significant DCT coefficients. A method based on motion vectors is presented in [14] and the other ones based on alteration of prediction loops are discussed in [116], [144] and [153]. Joint design of MD techniques with transport protocols is discussed in [90], [124] and in [123].

Chapter 3

Multi-State Video Coding

The two main problems associated with lossy packet networks such as the Internet are the limited bandwidth and packet losses. Video communication requires high compression and simultaneously high error resilience [13], [44]. Using retransmissions and/or forward error correction (FEC) are conventional approaches to overcome packet losses. In retransmission-based approaches a back-channel is necessary to inform the sender which packets are correctly received and which are lost. The disadvantage is the additional delay associated with the round-trip-time (RTT) between receiver-sender-receiver. If this delay is unacceptable for the application or if no back-channel is available, retransmissions cannot be employed.

FEC based approaches, on the other hand, add specialized inter-packet redundancy to the data to overcome losses. Reed Solomon block codes and Tornado codes are examples of FEC. FEC approaches are often combined with data interleaving to convert burst errors into isolated errors. While increasing the required bandwidth for transmission, FEC approaches are designed to protect against a predetermined amount of channel losses. While the data can be recovered perfectly when the losses are below the threshold, the data maybe be completely lost, if the threshold is exceeded. The problem with FEC is that network conditions such as packet loss are highly dynamic and there is limited knowledge about the current conditions. Limited knowledge on the other hand leads to inefficient FEC design.

Several other techniques based on scalable coding, prioritized data, combinations of ARQ and FEC, unequal error protection, multiple description coding are presented and discussed in [75], [146], [159], [156], [138] and [139]. The success depends mainly

on the requirements of the specific application.

In [4], a system composed of two subsystems is presented to provide reliable video communication over such networks. The two subsystems are the multi-state video encoder/decoder and a path diversity transmission system. Multi-state video coding (MSVC) codes the video into multiple independently decodable systems, each with its own prediction process and state. MSVC combats this way the problem of error propagation. The first advantage is, if one stream is lost, the other streams can still be decoded to produce acceptable video. Secondly, the correctly received streams provide bidirectional information from past and future which enables improved state recovery for the corrupted stream. MSVC is a kind of multiple description coding (MDC) with the novelty of the additional state recovery capability. The second subsystem, the path diversity transmission system, sends different subsets of packets over different paths. This way the packets proceed many paths instead of a single one and the end user observes average path behavior. The assumption is that the average path behavior provides better performance than the behavior of any individual random path. The probability that all of the multiple paths are simultaneously congested is less than the probability that a single path is congested and therefore the assumption is justified. The goal of MSVC is combining the two conflicting requirements of high compression and high error resilience.

The proposed system is independent from a feedback channel and is therefore suitable both for closed loop and open loop applications. Important open-loop applications include broadcast, multicast [95] and point to point applications. Because of the overhead of many responses, the back channel limits the scalability of one to many systems. For such systems a reliable video communication without relying on a back channel is especially important.

Conventional video compression standards use single-state system architecture. In [111], a work about joint source coding, transport processing and error concealment is processed. The single state corresponds to the previous coded frame. If the previous coded frame is lost the quality of the reconstructed subsequent frames is also affected until the prediction is refreshed, i.e. the state is reinitialized. Since there are multiple independently decodable streams in MSVC, if one state is corrupted the other states remain accurate. The corresponding streams can be decoded and also be used to recover the stream with the lost state. The novelty of MSVC is the idea of using the

data from multiple streams to recover the lost state. Redundancy between frames in different streams are exploited to improve the recovery of the lost frames.

3.1 Encoder Portion of the System

In the two-state MSVC, the input video is partitioned into two subsequences of frames (even and odd) which are coded into two separate bitstreams. Each stream has a different prediction loop [42], [49] and a different state and is independently decodable from each other. The encoder consists of two separate conventional encoders. The disadvantage is that a higher bitrate is required to code the frames in multiple streams instead of a single one. Since the distance between neighboring frames in the same stream increases, the prediction gain decreases. The idea is similar to Video Redundancy Coding (VRC) presented in [159] with the difference that there are no redundantly coded frames. To ensure independent losses for the generated streams, the two descriptions should be sent over different channels undergoing independent error effects.

3.2 Decoder Portion of the System

The decoder can consist of two separate decoders. If both the streams are received error-free, the independently decoded even and odd streams can be interleaved to produce the final stream for display. On the other hand, if there is an error, the state for that stream is incorrect and there will be error propagation while further decoding it. The other independently decodable stream on the other hand can still be decoded error-free to produce usable video. In this case, the video is recovered at half its original frame rate. Although the frame rate is reduced, there are no other distortions in the produced video. If we have a single-state stream instead, the lost frame is either replaced by the previous decoded frame or it is reconstructed using some kind of concealment. If there are many frames before the next I-frame, the overall distortion can become significant.

The advantage of MSVC is that it provides improved error concealment and enables improved state recovery of the corrupted stream. Whereas single-state approaches only have access to previous frames to use in error concealment, MSVC can

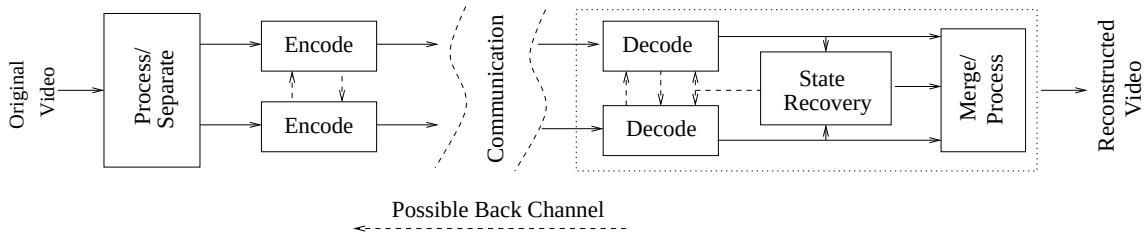


Figure 3.1: Block Diagram of the MSVC System

access both the previous and future frames. Using both previous and future frames can greatly help in recovering the corrupted stream and restore the video to its full frame rate. The lost frame can often be estimated with sufficient accuracy to be used as a reference for predicting other frames in that stream. This way, the corrupted stream can be recovered before reaching the point of next resynchronization.

Another advantage is that multiple states can help to estimate the quality of state recovery, i.e. the recovered corrupted stream can be used to estimate the known correctly received stream and the recovery quality is estimated by the accuracy of the match. Knowledge of the quality can be incorporated in several ways to the reconstruction process. The decoder may repeat the last correctly decoded frame and wait for the next resynch or continue to decode and display all the frames depending on whether the quality is acceptable or not. Figure 3.1 shows the block diagram of the MSVC approach including both the encoder and decoder parts.

3.3 State Recovery

For state recovery, the lost frame is to be estimated using both previous and future frames in the sequence. The goal of state recovery is to produce an accurate estimate of the coded frame so that it can be used to form an accurate prediction of the subsequent frames. There are different methods for estimating the lost frame. The simplest one is replacing the lost frame by a correctly decoded frame. A more complex method is interpolating the previous and next frame [89], [109]. Even the motion field can be computed across a subset of correctly decoded past and future frames from the corrupted and uncorrupted streams. The interpolation can also consider the covered and uncovered areas within the frame to enhance the interpolation accuracy.

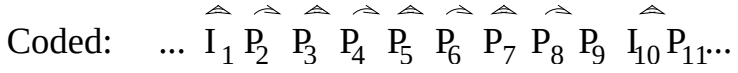


Figure 3.2: Prediction Process in Single-State System

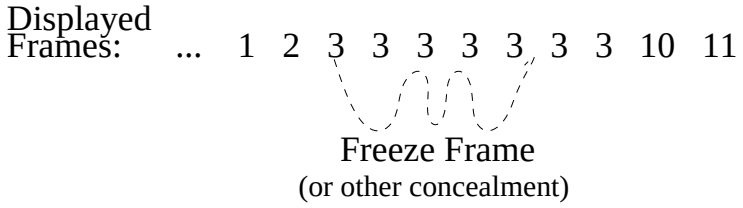


Figure 3.3: Signal Reconstruction in Single-State System

The concealment is then a linear or nonlinear filtering along the motion trajectory. In case of MSVC, coded information within the bitstream can be used to reduce the complexity of concealment, i.e. state recovery. Figure 3.2 shows the prediction process in a single-state system and Figure 3.3 the sequence reconstruction when the fourth frame in the sequence is lost. The last frame received, in this case the third frame, is repeated until the next intra frame update, which occurs on the tenth frame. Similarly, Figure 3.4 shows the prediction and signal reconstruction in case of losses for the MSVC system. Figure 3.5, on the other hand depicts the error concealment in MSVC.

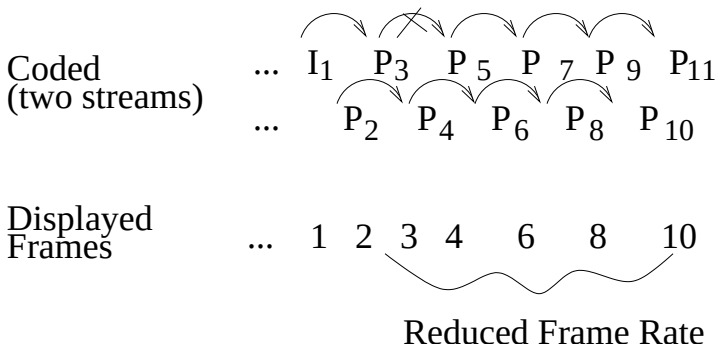


Figure 3.4: Prediction Process and Signal Reconstruction in MSVC

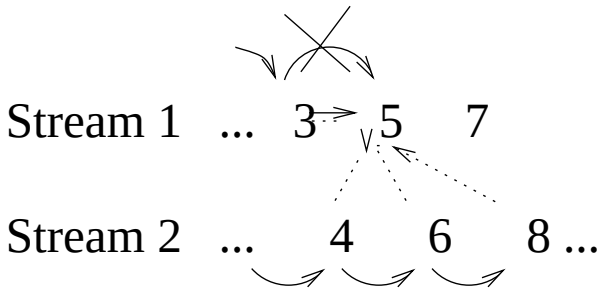


Figure 3.5: Error Concealment in MSVC

3.4 Approach 1

Approach 1 is the original method presented in [4] and consists of the following steps:

1. The video is encoded into multiple (e.g. 2) independently decodable packet streams
2. Each packet stream is transmitted over a different path over the packet network
3. Multiple packet streams containing some losses are received
4. MSVC is used to decode the streams by applying state recovery to recover the lost states and enable further decoding on corresponding streams.

In the experiments, we produce two independently decodable streams corresponding to two states. Each stream is then sent over a different path. The existence of an ideal path diversity system is assumed, the architecture is irrelevant for further discussion. The ideal path diversity system provides two paths with independent losses according to its definition.

All sequences used in the experiments are in QCIF format (144 x 176 pixels/f) and coded at a frame rate of 30f/s. The multi-state video codec is developed on top of the H.264 codec (version 9.0, H.26L). The block diagrams of H.264 encoder and decoder are shown in Figures 3.6 and 3.7. “ME” in the figures stands for Motion Estimation and “MC” for Motion Compensation. “QP” is the quantization step size, “INTRA” and “INTER” are the coding modes: in intra-coding no prediction is performed from other pictures, in inter-coding on the other hand a prediction signal is formed from the reference frame using the motion information estimated in the previous step. The

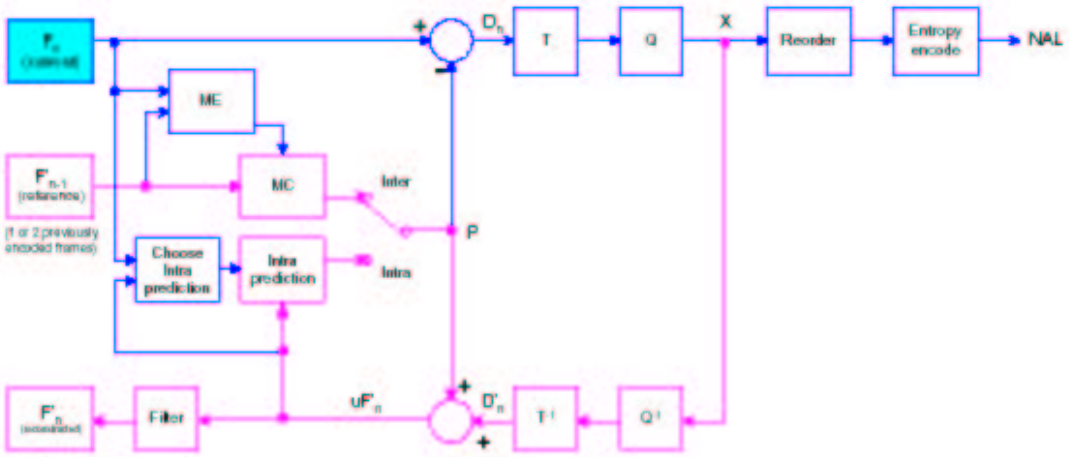


Figure 3.6: H.264 Encoder, Reference [94]

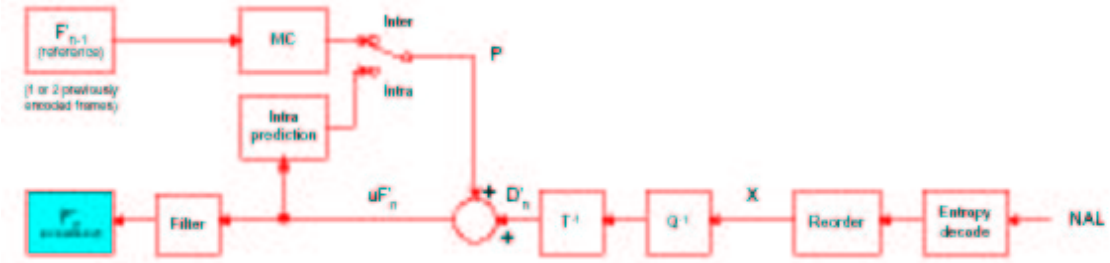


Figure 3.7: H.264 Decoder, Reference [94]

sequences are coded into two streams (containing the even and odd frames) at 15 f/s with a constant quantization step size (QP) yielding nearly the same PSNR for all frames. We assume for simplicity, that each frame coded as intra (I) or P-Frame is sent in a single packet. If a packet is lost we assume that all data corresponding to that frame is lost. If a frame is lost, the concealment is performed through a motion controlled block based interpolation. To do this, the motion vectors of the next frame on the other stream are used. There are definitely better concealment methods, such as MCinterp which is also applied in [4], but we have chosen to use the motion vectors already coded and received in the stream. Moreover the complexity of this block-based concealment approach is relatively low.

3.4.1 One of the channels is lossless

The strength of MSVC is depicted in Figure 3.8, where it is compared to SSVC. MSVC uses two paths, on one of the paths occurs a single packet loss containing frame 17. The path over which the SSVC packets are transmitted undergoes also a single packet loss corresponding to frame 17. In MSVC case, the lost frame is recovered through a motion based interpolation of frames 16 and 18. The motion vectors directed from 18 to frame 16 are employed for the interpolation. In case of SSVC, on the other hand, the last frame 16 was repeated to replace frame 17. The PSNR drop for Foreman and MSVC is about 1 dB, quite small compared to the huge drop of 8 dB for SSVC transmission of the same sequence. In MSVC, each second frame is unaffected from the loss whereas in SSVC all of the subsequent frames undergo error propagation caused by the frame repetition. In contrary to the Foreman sequence, which is a high motion sequence, Akiyo is a low motion sequence. For low motion sequences, frame concealment through frame interpolation as well as frame repetition is much easier and therefore PSNR drop is much smaller. Figure 3.9 shows the comparison of MSVC and SSVC for a single loss (frame 17) for Akiyo. The SSVC frame PSNR drop is much larger due to frame repetition (about 2 dB) as compared to 0.5 dB drop due to MSVC frame interpolation. But since SSVC has a larger average coded frame PSNR at the same bitrate, the PSNR of the frame succeeding the lost frame is still larger than in case of MSVC. On the other hand, we know that PSNR differences between neighboring frames which are larger than 0.5 dB are visually disturbing for the observer. In this aspect, MSVC has a smooth PSNR characteristic, although the drop of 2 dB for a slow motion sequence will be detected visually. The initial PSNR gap between SSVC and MSVC is due to the increased time distance between neighboring frames on the same thread of MSVC: as the distance increases the prediction gain decreases, i.e. at the same coding bitrate the average PSNR decreases. Moreover, for the low motion sequence, Akiyo, which has a large coding gain, the cost of I frames is extra high compared to P frames. Since our MSVC implementation uses one I frame (the first one) for each stream, the PSNR difference between SSVC and MSVC for the Akiyo sequence is about 2 dB. Using the same intra coded reference frame for both streams would lower this difference, which is not further explored here.

Figures 3.10 and 3.11, on the other hand, show the result of a similar experiment

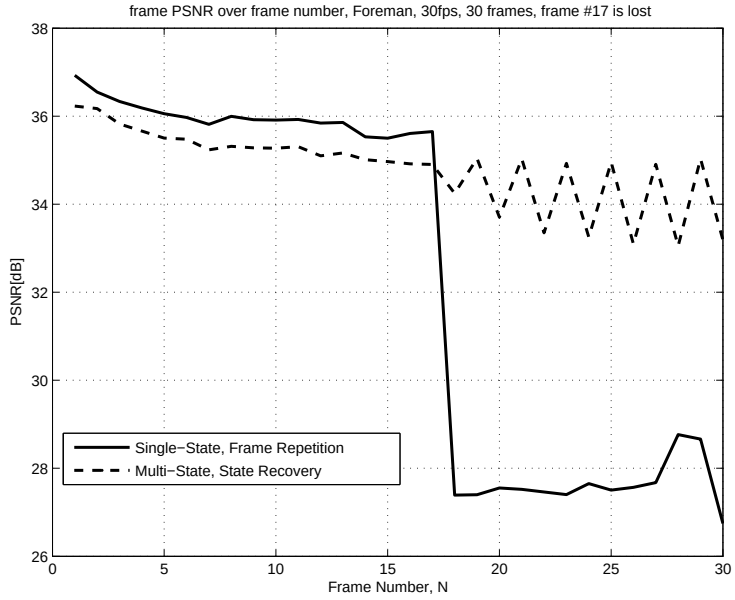


Figure 3.8: Frame PSNR over Frame Number, Approach 1, Foreman sequence, frame #17 is lost

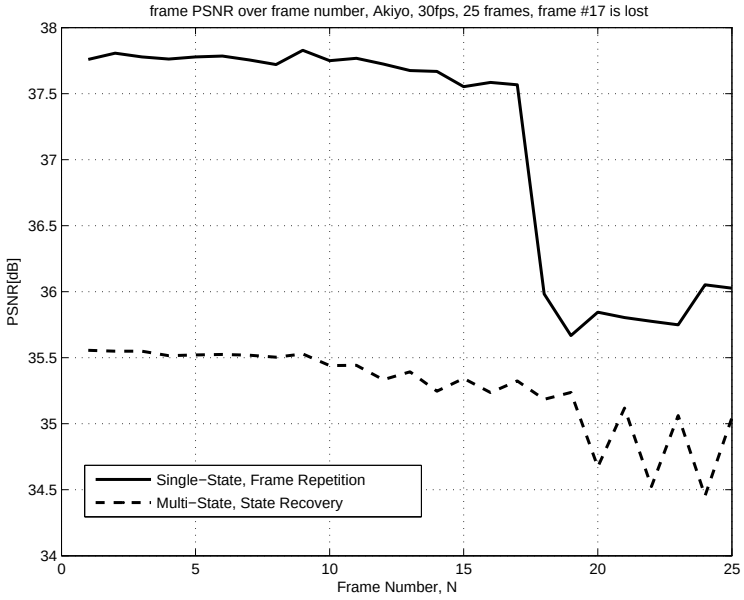


Figure 3.9: Frame PSNR over Frame Number, Approach 1, Akiyo sequence, frame #17 is lost

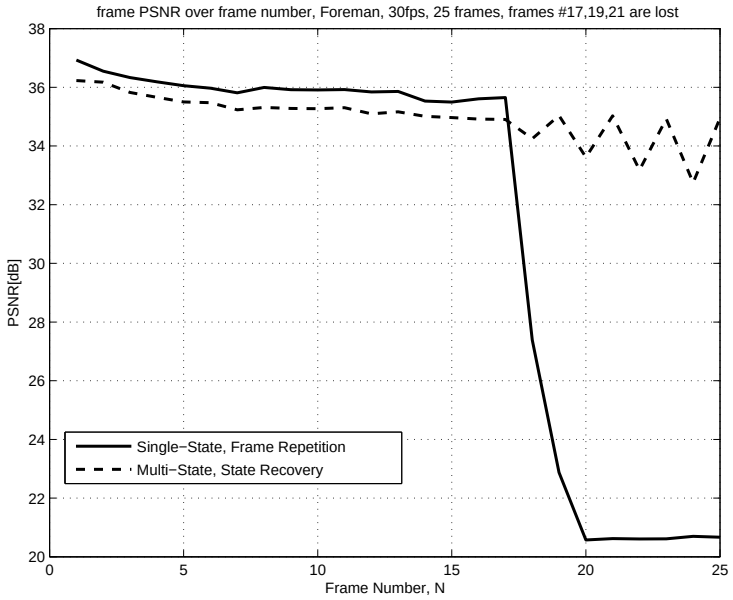


Figure 3.10: Frame PSNR over Frame Number, Approach 1, Foreman sequence, frames #17, 19 and 21 are lost

for Foreman and Akiyo respectively. Instead of a single loss, three consecutive losses (burst losses) occur on the same stream corresponding to frames 17, 19 and 21. The overall PSNR drop for SSVC is 14 dB for Foreman and 6 dB for Akiyo. The drops for MSVC, on the other hand, are 1 dB for Foreman and less than 0.5 dB for Akiyo. Despite the initial advantage of SSVC, after experiencing three losses, the PSNR of the consecutive frames are about 3 dB lower than in case of MSVC. As seen here, the superiority of MSVC is more obvious in case of burst errors.

Figures 3.13 and 3.12 show the comparison of SSVC and MSVC for the case of an burst error (3 consecutive losses) followed by two single errors separated by 0.133 and 0.4 seconds apart from each other. The frame PSNR course over 50 frames is depicted so that the error propagation and the recovery from errors in time dependence can be observed. The maximal drop in PSNR for Foreman at SSVC is about 15 dB, the recovery from errors is very slow. The recovery for Akiyo is much faster where the maximal PSNR drop is 6 dB. We see that even after several frames after the last error, the PSNR level of SSVC did not reach the PSNR level of MSVC though the initial PSNR advantage.

Next, we consider the case where 10 consecutive packets (=10 frames) are lost

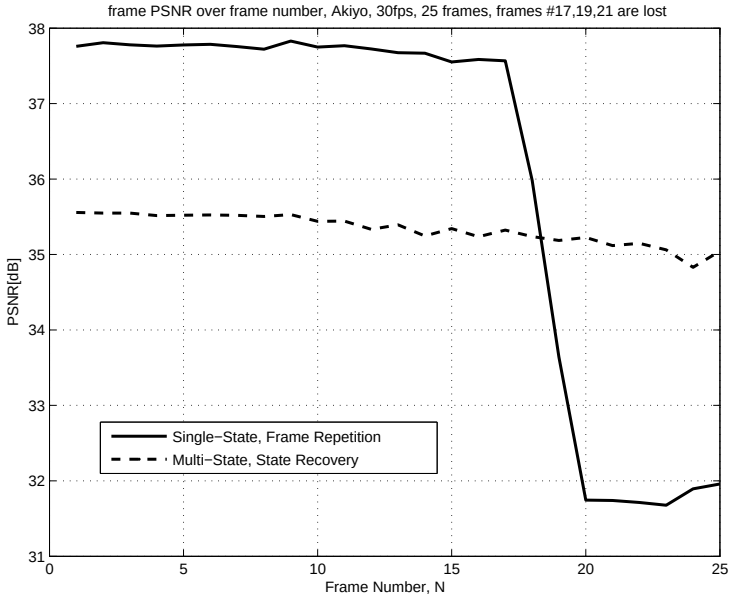


Figure 3.11: Frame PSNR over Frame Number, Approach 1, Akiyo sequence, frames #17, 19 and 21 are lost

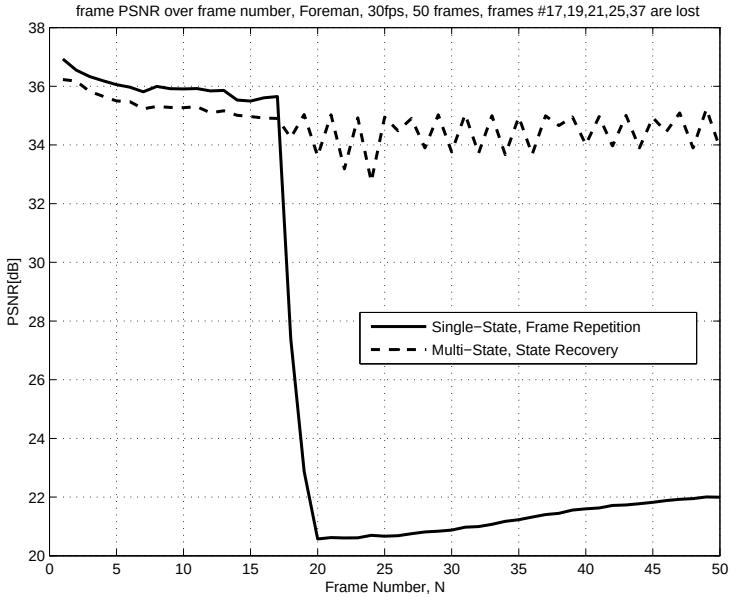


Figure 3.12: Frame PSNR over Frame Number, Approach 1, Foreman sequence, frames #17, 19 ,21, 25 and 37 are lost

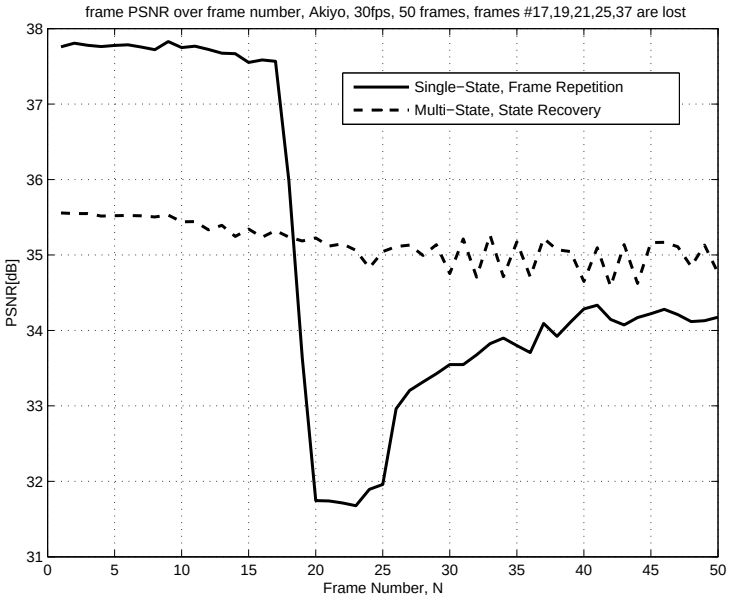


Figure 3.13: Frame PSNR over Frame Number, Approach 1, Akiyo sequence, frames #17, 19 ,21, 25 and 37 are lost

starting with the frame number 61 both for MSVC and SSVC. We show the reconstruction results demonstrating two effects: 1- frame loss and 2- error propagation. Figure 3.14 depicts the original frames # 70, 71 and 72 from the sequence Foreman. The lost and by state recovery (MSVC) reconstructed frame # 71 and the adjacent frames # 70 and # 72 from the loss-free stream are given in Figure 3.15. The loss of ten consecutive odd frames corresponds to a bursty loss on the odd thread. Figure 3.16 shows for SSVC the reconstructed frames # 70, 71 and 72. Frame # 70 is the repetition of the last received frame # 60. Figures # 71 and 72 show the effect of propagation after frame freeze. The effect of error propagation becomes more obvious on the proceeding frames after the burst loss. Figure 3.17 shows the original frames # 79, 80 and 81. The same frames for SSVC are given in 3.19 showing the error accumulation due to error propagation. On the other hand, 3.18 gives the results for MSVC and the same frames where frame # 80 was transmitted on the lossless channel. Frame # 79 is lost and reconstructed through state recovery and frame # 81 contains the error due to frame interpolation after propagation. Not only is the reconstruction error is smaller for MSVC but also the recovery from losses and the corresponding errors is much faster.

Figure 3.14: Foreman, original, Frames #70, #71 and #72

Figure 3.15: Foreman, MSVC, Frames #70, #71 and #72

Figure 3.16: Foreman, SSVC, Frames #70, #71 and #72

Figure 3.17: Foreman, original, Frames #79, #80 and #81

Figure 3.18: Foreman, MSVC, Frames #79, #80 and #81

Figure 3.19: Foreman, SSVC, Frames #79, #80 and #81

	avg. PSNR [dB]	Bitrate [kbits/s]	QP
F.	35.87	292.05	16
A.	34.24	16.30	22
C.	35.89	19.50	22
M&D.	35.86	77.81	17
S.	35.01	69.46	17

Table 3.1: Balanced MSVC, Coding Parameters

Similarly, Figures 3.20, 3.21, 3.22, 3.23 and 3.24 show further experimental results where the even stream is received lossless ($p_1 = 0$) and the odd stream lossy ($p_2 \neq 0$). We tested specially the loss rates of 5%, 10% and 20% for the lossy stream. The figures show the averaged frame PSNR's over the frame number (200 Frames) for five sequences: Foreman, Akiyo, Claire, Mother & Daughter and Salesman. The sequences are coded without intra frame or block updates and are decoded for 100 different randomly generated loss patterns with the given loss rates p_1 and p_2. The frame PSNR's are averaged over the 100 measurements. We coded even and odd sequences with the same quantization step size yielding the same average frame PSNR and bitrate for the two streams. But different sequences have different average bitrates and frame PSNR's which are listed in Table 3.1. "F." denotes the sequence Foreman whereas "A." stands for Akiyo, "C." for Claire, "M&D." for Mother & Daughter and "S." for Salesman.

Figure 3.20 shows the PSNR characteristics for Foreman. As expected when $p_2 = 0.2$ the average PSNR of the lossy stream is the smallest and it is largest if p_2 is as small as possible. But while exploring the remaining figures 3.21, 3.22, 3.23 and 3.24, an interesting effect is noticed: after a certain frame number the course changes, i.e. as p_2 increases the average frame PSNR increases. This effect is highly dependent on the state recovery mechanism chosen. For lower motion sequences as Akiyo, Claire, even Mother & Daughter and Salesman, the interpolation yields very good results. The error propagated to the consecutive frames after frame interpolation is larger than the interpolation error itself. If another loss occurs following the first loss, the PSNR level is increased through a further frame interpolation. If the sequence has high motion on the other hand, the interpolation error is larger than the error propagation. In other words, the results are highly related to the interpolation algorithm. We use here a very simple block based interpolation method, do not perform anything

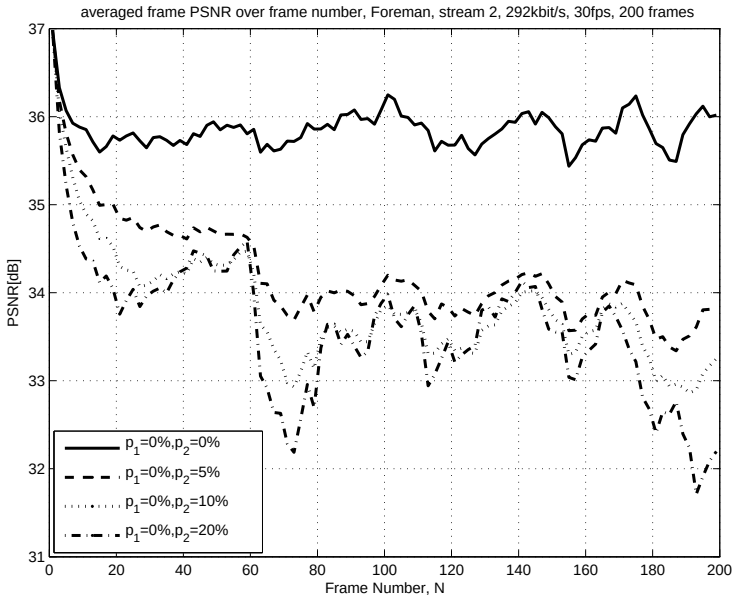


Figure 3.20: PSNR over Frame Number, Approach 1, Foreman sequence, first stream received lossless

special for occlusions or exposures and use the motion information available from the lossless received stream. A pixel based interpolation algorithm like MCInterp could generate better interpolation results affecting our test results more in favor of interpolations. A much simpler frame concealment method such as averaging the two neighboring frames from the received stream, on the other hand, would worsen the interpolation results and favor the reception of packets instead of interpolations. Another observation is that the general form of frame PSNR over frame number curve changes only slightly with the increasing loss rate. This is in accordance with our expectations. The frames needing more bitrate have smaller PSNR values when coded with the same quantization step size. They are also the ones which differ most from their reference frames and therefore the ones which are difficult to reconstruct through error concealment.

Figures 3.25, 3.26, 3.27, 3.28 and 3.29 show a comparison of frame PSNR's of the interleaved reconstructed sequence over frame number depending on the loss rate of the second stream p_2. Only the first 20 frames are depicted on the figures. As the frame number increases, lossy transmissions of the second stream yield better results for low motion sequences. Therefore we have focused on the first part of

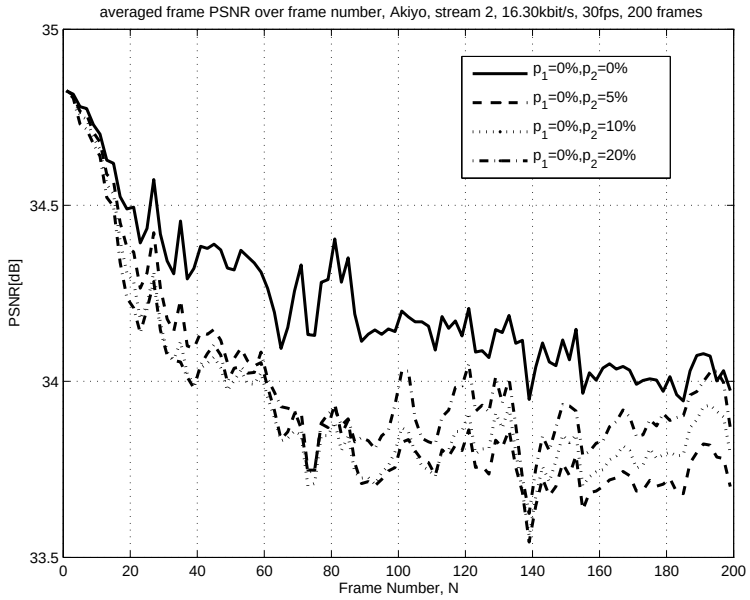


Figure 3.21: PSNR over Frame Number, Approach 1, Akiyo sequence, first stream received lossless

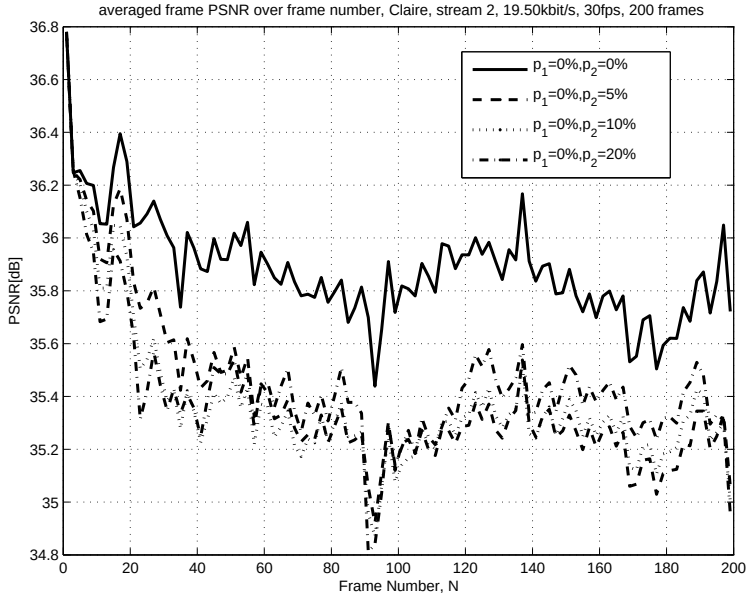


Figure 3.22: PSNR over Frame Number, Approach 1, Claire sequence, first stream received lossless

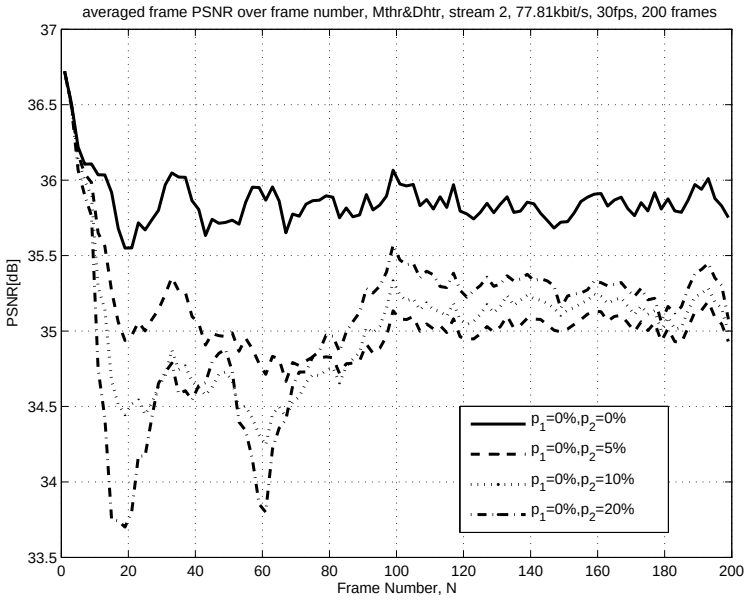


Figure 3.23: PSNR over Frame Number, Approach 1, Mother & Daughter sequence, first stream received lossless

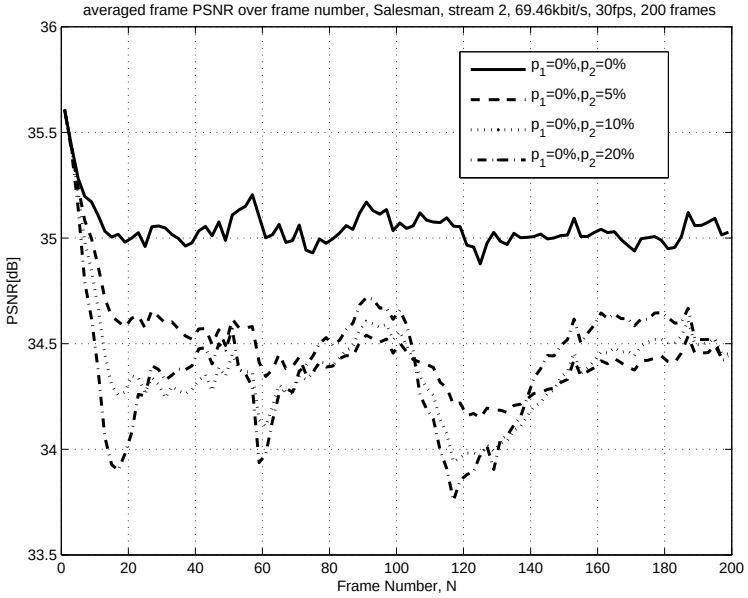


Figure 3.24: PSNR over Frame Number, Approach 1, Salesman sequence, first stream received lossless

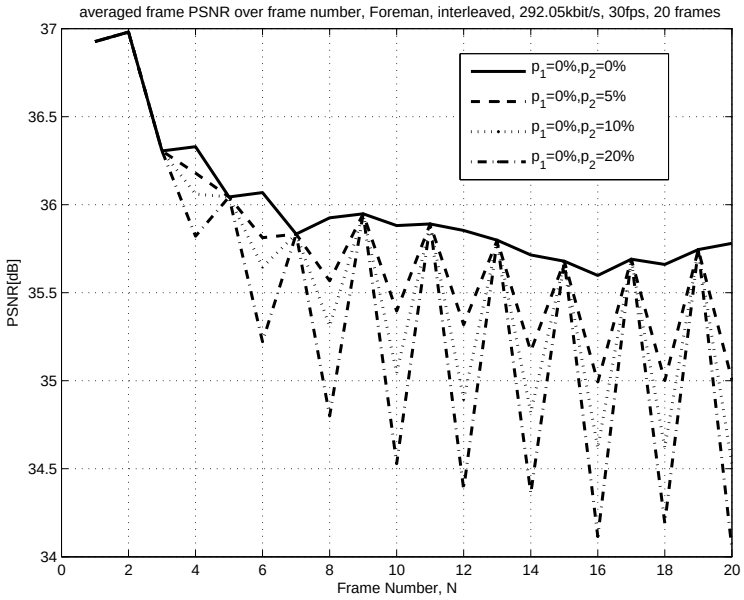


Figure 3.25: PSNR over Frame Number, Approach 1, Foreman sequence, interleaved, first stream received lossless

the sequence where higher loss rates yield smaller average frame PSNR values. The presented values are averaged over 100 randomly generated loss patterns with given p_1 and p_2. The PSNR difference between neighboring frames is at most 1.5 dB for the high motion Foreman sequence and about 0.4 dB for Akiyo. Akiyo and Claire can be classified as very low motion sequences whereas Mother & Daughter and Salesman are moderately high motion sequences as verified by the frame PSNR differences between subsequences after state recovery and interleaving.

3.4.2 Both channels are lossy

In the previous subsection it was shown that if one of the channels is always received, employing MSVC is advantageous to recover from single as well as and more importantly from burst errors and outages. Here we investigate the case of applying MSVC to a sequence of 200 frames (about 6.7 seconds) without any intra updates in between. We assume also that both channels are lossy with uniform and independent loss probabilities p_1 and p_2. Figures 3.30, 3.31,3.32,3.33 and 3.34 show the average PSNR over frame number when $p_1 = 0.05$ for Foreman, Akiyo, Claire, Mother & Daughter and Salesman respectively. The coding parameters are given in Table 3.1.

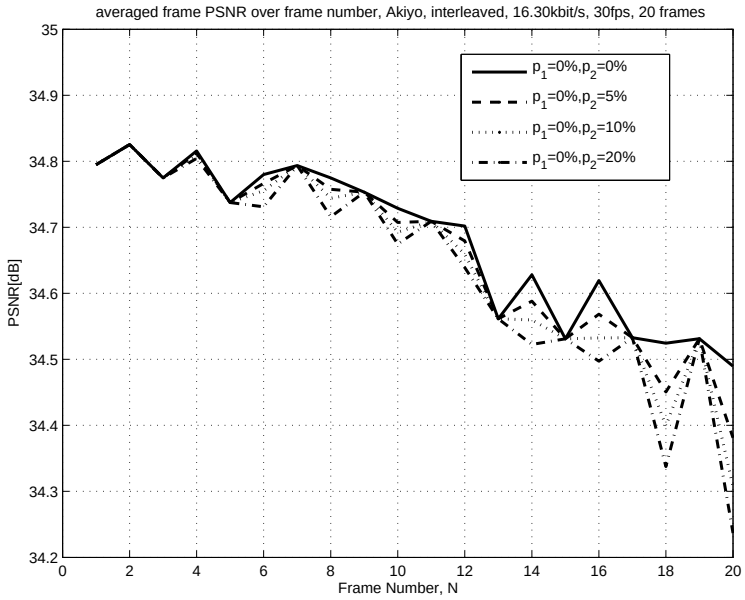


Figure 3.26: PSNR over Frame Number, Approach 1, Akiyo sequence, interleaved, first stream received lossless

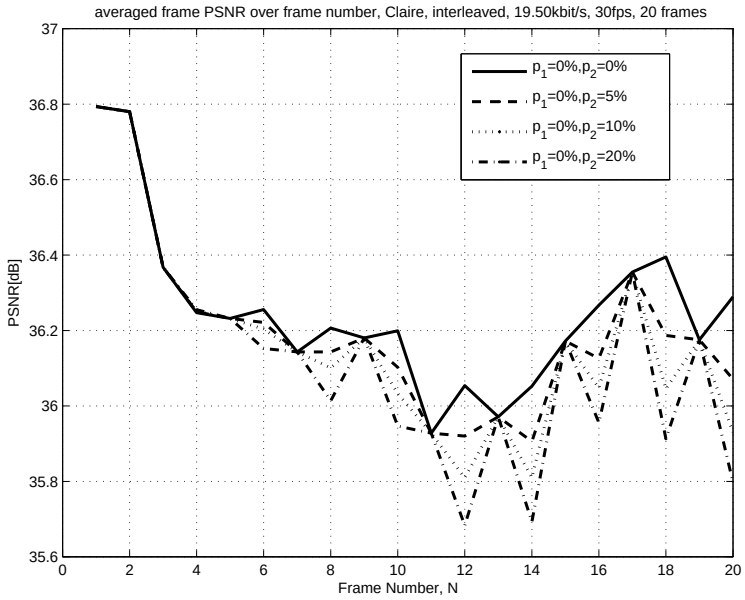


Figure 3.27: PSNR over Frame Number, Approach 1, Claire sequence, interleaved, first stream received lossless

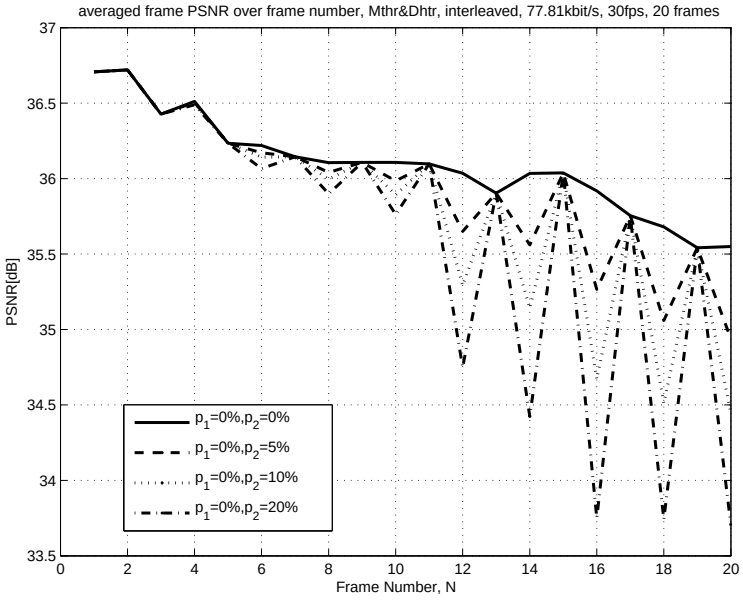


Figure 3.28: PSNR over Frame Number, Approach 1, Mother & Daughter sequence, interleaved, first stream received lossless

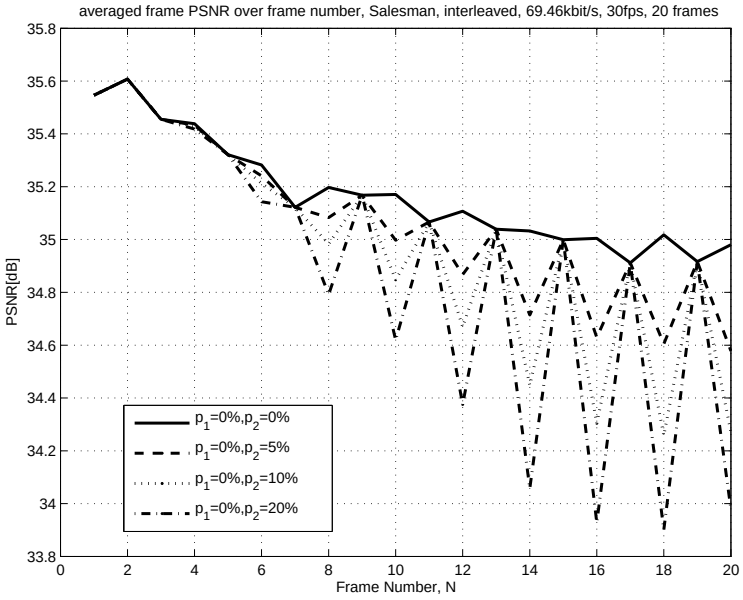


Figure 3.29: PSNR over Frame Number, Approach 1, Salesman sequence, interleaved, first stream received lossless

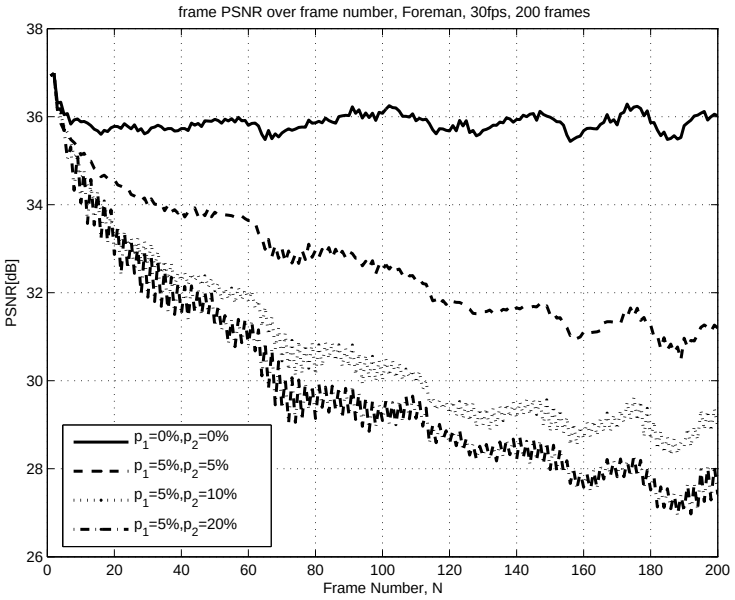


Figure 3.30: PSNR over Frame Number, Approach 1, Foreman sequence, interleaved, both streams received lossy

When $p_1 \neq p_2$ the flickers (PSNR differences between neighboring frames) increase. If p_1 is fixed the changes in p_2 affect the frame PSNR relatively small, i.e. at most 2 dB (between 5% and 20%) for Foreman and 1 dB for Akiyo. As the loss rates of both channels increase the interaction of the streams in form of interpolations and state recovery increases so that both channels get affected if losses occur on any of them. This is also the reason why flickers become smaller as the frame rate increases.

Figures 3.35, 3.36, 3.37, 3.38 and 3.39 show the case p_1 and p_2 varies between 0.01 and 0.02. The $p_1 = 0, p_2 = 0$ case is depicted as reference only. The figures correspond to transmission of 200 Frames averaged over 100 randomly generated loss patterns for sequences Foreman, Akiyo, Claire, Mother & Daughter and Salesman. The observation is as above: If both streams are lossy, increasing the loss rate of one of them decreases the average PSNR of both streams. This decrease is largest for Foreman and smallest for Akiyo.

Next we fix the loss rate of the second stream as $p_2 = 0.2$ and vary only p_1 between 0 and 0.2 for the five sequences respectively. The results are given in Figures 3.40, 3.41, 3.42, 3.43 and 3.44. Increasing p_1 from 0 to 0.05 decreases the average PSNR drastically over all frames in both streams (around 7 dB at the 200^{th} Frame). The first

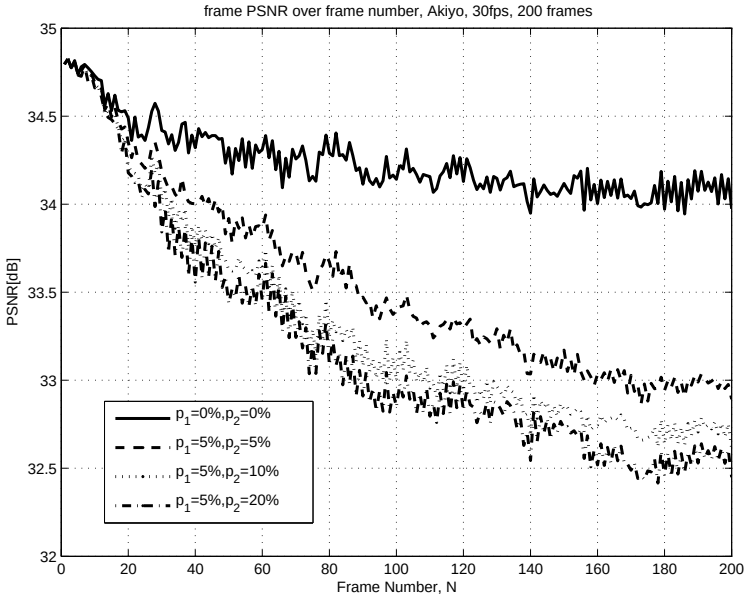


Figure 3.31: PSNR over Frame Number, Approach 1, Akiyo sequence, interleaved, both streams received lossy

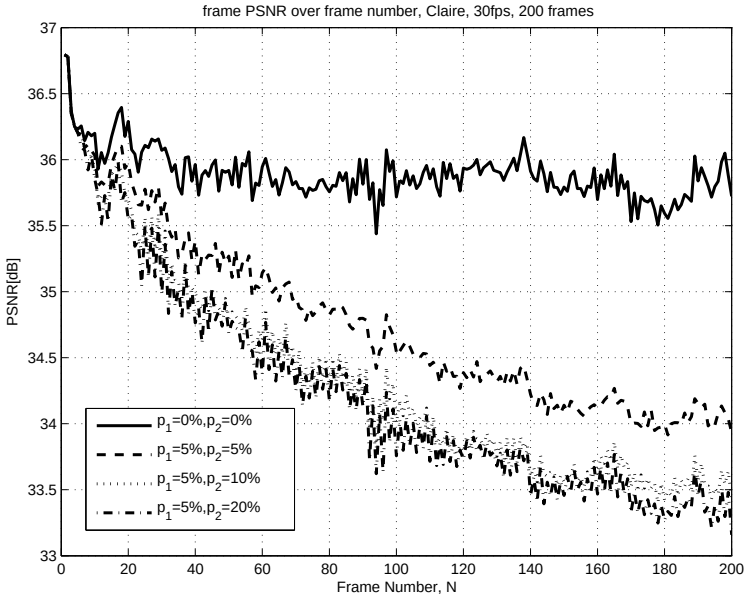


Figure 3.32: PSNR over Frame Number, Approach 1, Claire sequence, interleaved, both streams received lossy

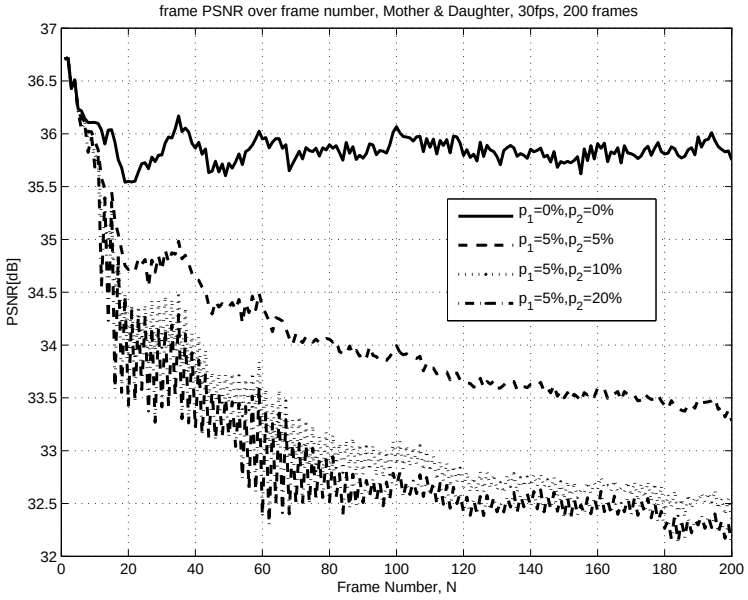


Figure 3.33: PSNR over Frame Number, Approach 1, Mother & Daughter sequence, interleaved, both streams received lossy

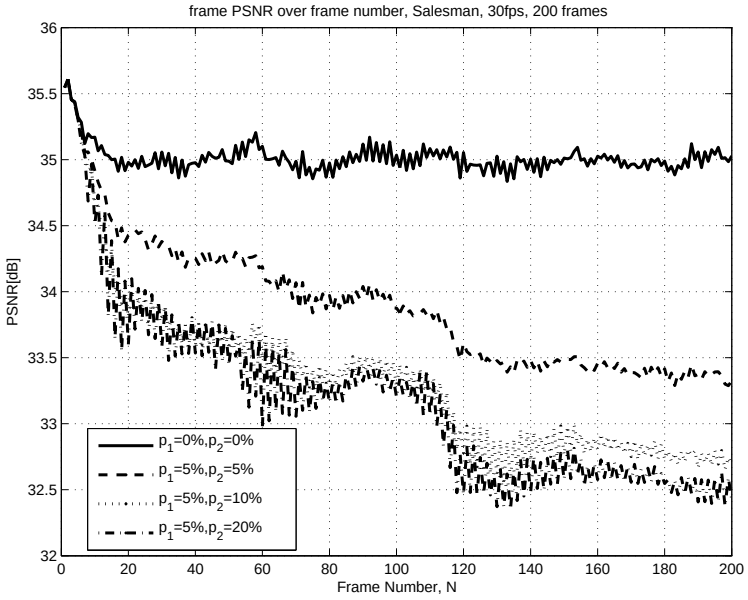


Figure 3.34: PSNR over Frame Number, Approach 1, Salesman, interleaved, both streams received lossy

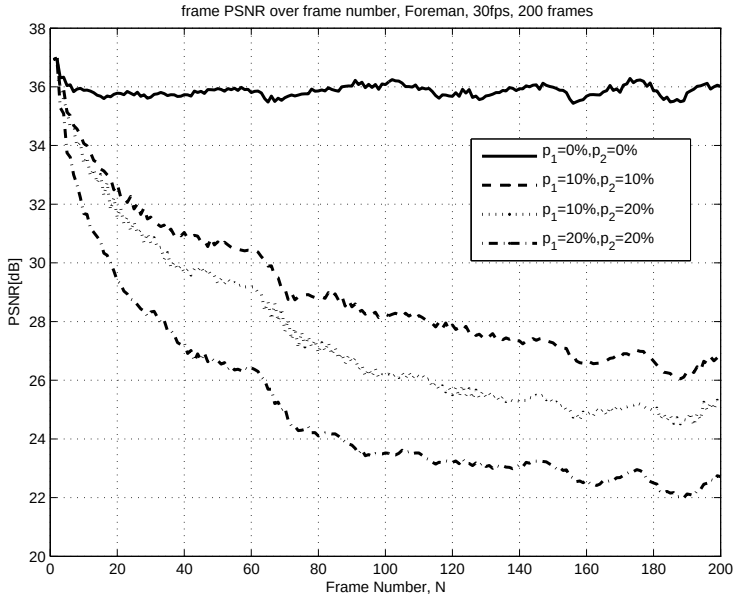


Figure 3.35: PSNR over Frame Number, Approach 1, Foreman sequence, interleaved, both streams received lossy

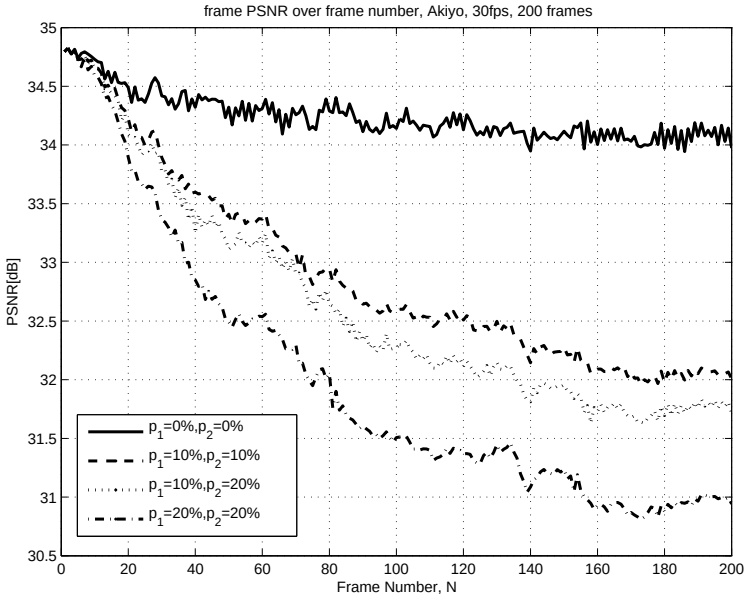


Figure 3.36: PSNR over Frame Number, Approach 1, Akiyo sequence, interleaved, both streams received lossy

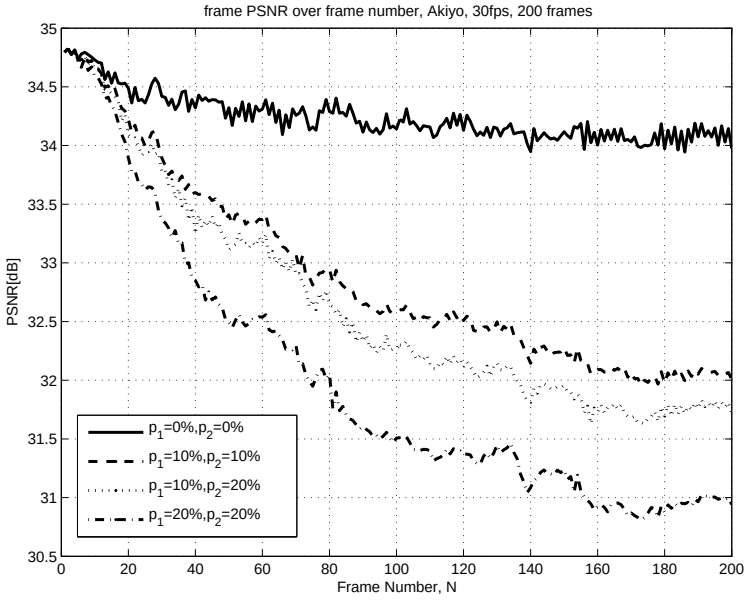


Figure 3.37: PSNR over Frame Number, Approach 1, Claire sequence, interleaved, both streams received lossy

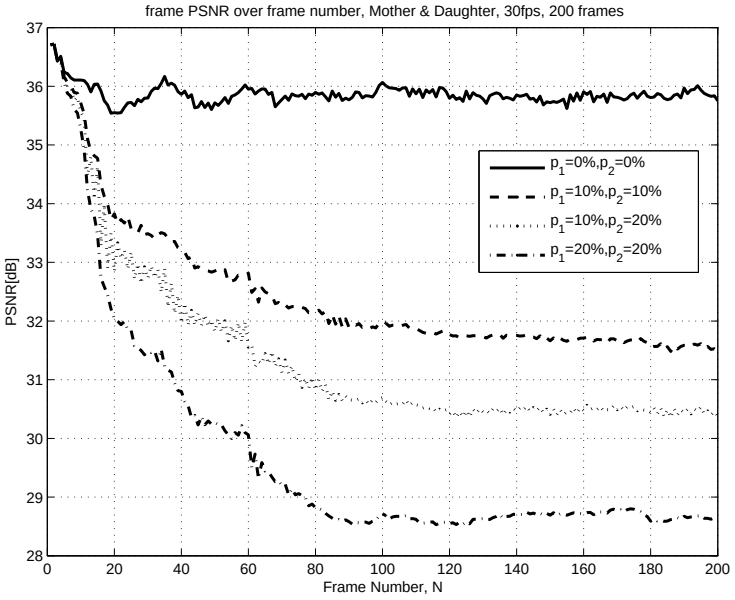


Figure 3.38: PSNR over Frame Number, Approach 1, Mother & Daughter sequence, interleaved, both streams received lossy

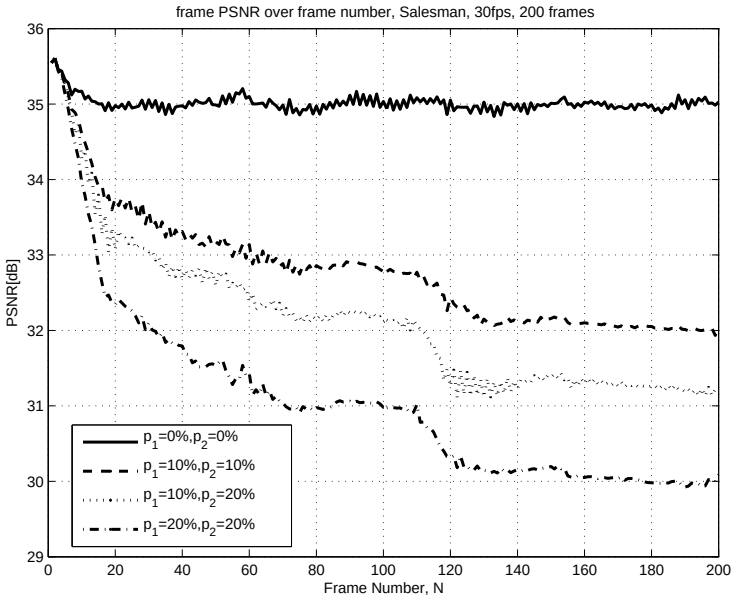


Figure 3.39: PSNR over Frame Number, Approach 1, Salesman sequence, interleaved, both streams received lossy

stream is no more in position to recover the second stream from errors. Since both streams are interacting, as the frame number increases, the average frame PSNR's of the two streams get closer, although the loss rates are different. Increasing p_1 from 0.05 to 0.1 or from 0.1 to 0.2 does not cause such a drastic decrease but about 2 dB at the 200^{th} Frame of Foreman. The PSNR differences are much lower for low motion sequences. Another point of interest is the PSNR differences between neighboring frames. As depicted in the figures the differences are largest when the first stream is lossless and the second one is lossy with $p_2 = 0.2$. The flickers increase with increasing p_2.

3.4.3 MSVC with intra-updates

Applying intra-updates is also an error resilience technique. We will combine both error resilience techniques, MSVC and intra-updates to investigate their performance at various combinations and the same bitrate. Two intra-update mechanisms will be tested: 1-intra-updates of frames, 2-intra-updates of GOB's (group of blocks). A GOB is equivalent to a line of macroblocks at a QCIF frame. A macroblock consists of 16x16 pixels and a GOB of 16x176 pixels. Periodic intra-coding of whole frames

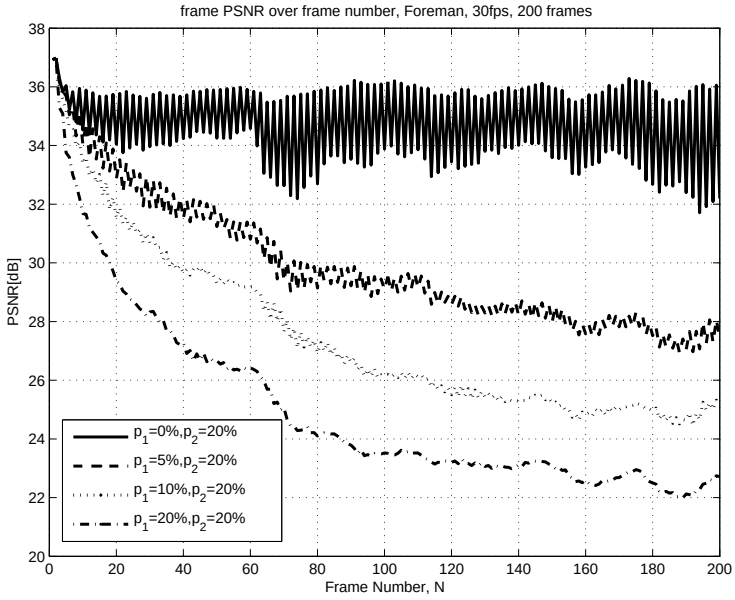


Figure 3.40: PSNR over Frame Number, Approach 1, Foreman sequence, interleaved, both streams received lossy

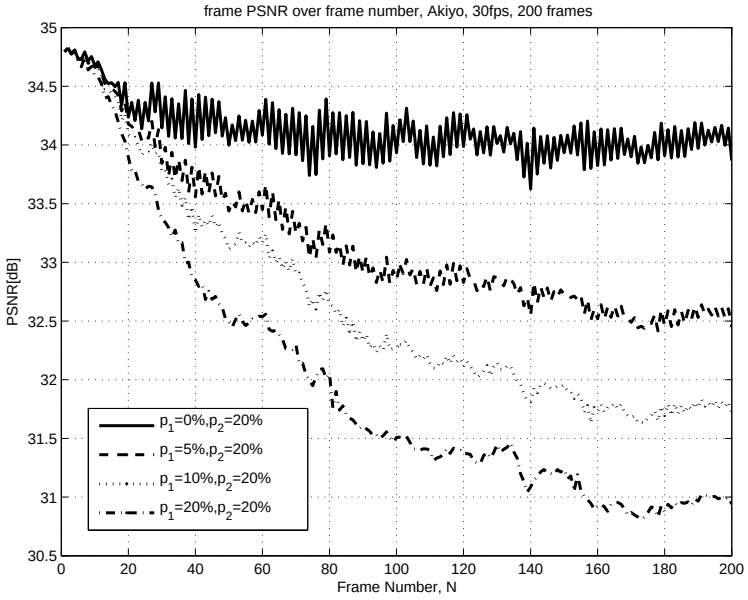


Figure 3.41: PSNR over Frame Number, Approach 1, Akiyo sequence, interleaved, both streams received lossy

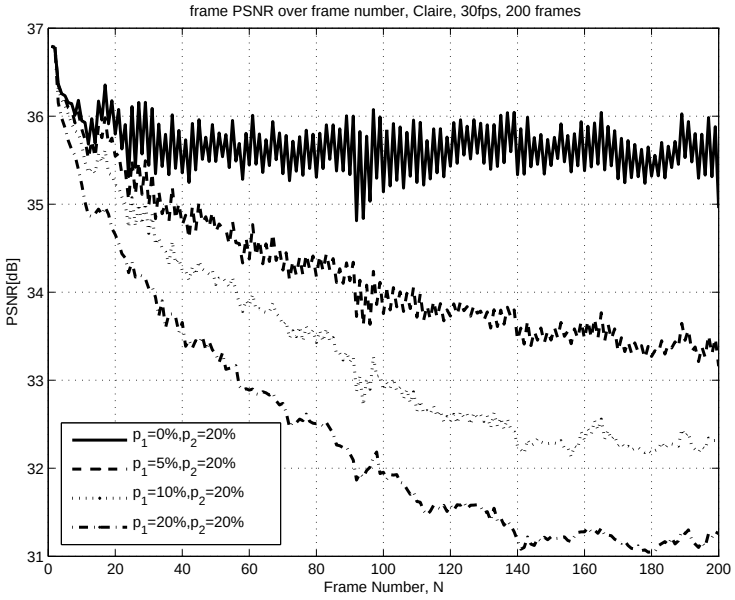


Figure 3.42: PSNR over Frame Number, Approach 1, Claire sequence, interleaved, both streams received lossy

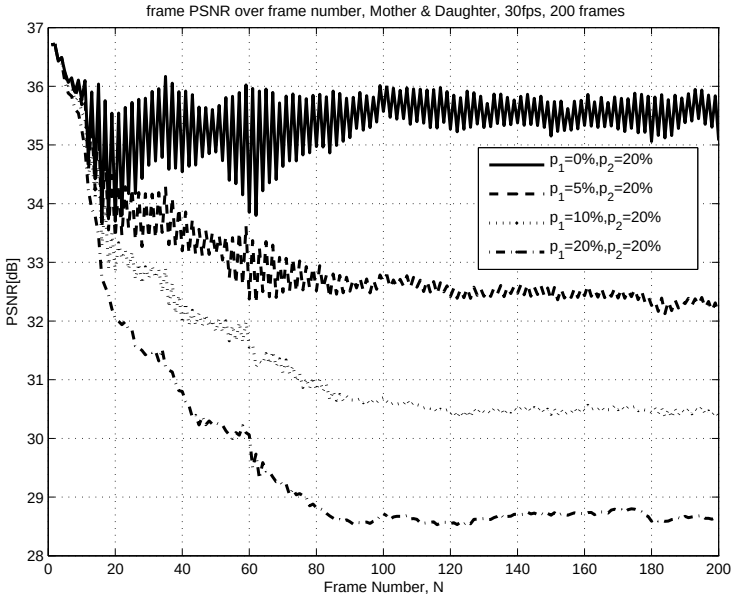


Figure 3.43: PSNR over Frame Number, Approach 1, Mother & Daughter sequence, interleaved, both streams received lossy

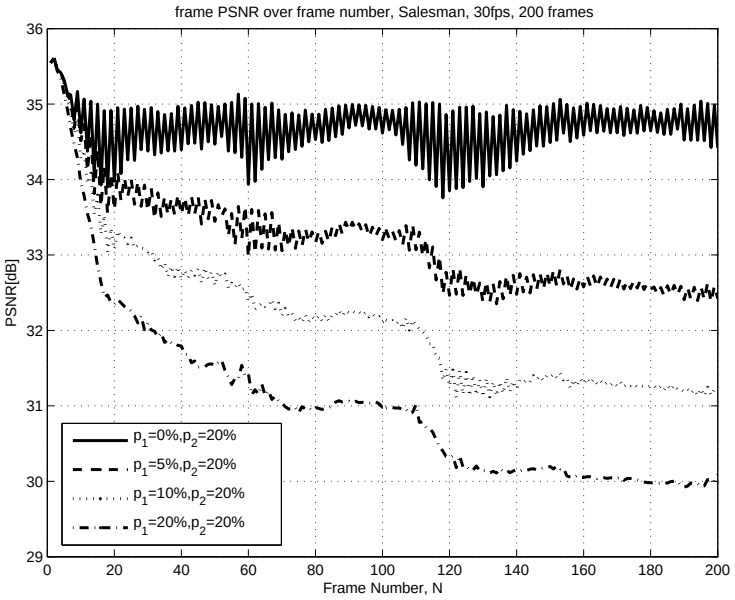


Figure 3.44: PSNR over Frame Number, Approach 1, Salesman sequence, interleaved, both streams received lossy

is investigated in [140], intra coding of contiguous blocks in [111] and intra coding of random blocks in [23]. These methods use a heuristic relationship between the packet loss rate and the refresh frequency, but apply intra-coding uniformly to all the regions of the frame. Reference [68] discusses the content adaptive method which applies frequent intra-update to regions that undergo significant changes . In [92] and [131], on the other hand, methods are presented that apply intra-update to regions where a rough estimate of decoder exceeds a given threshold. An early proposal of mode selection based on a RD framework to combat packet loss is given in [70]. In [23] and [158], the encoder takes into account the effects of error concealment in RD based mode selection. Error concealment is incorporated in computation of decoder distortion at the encoder in [66], [131] and [74]. Reference [104] presents a more general discussion of RD frameworks that incorporate channel error. A simple rate control scheme for rate distortion optimized video coding in an error-free environment can be found [135]. References [160] and [18], on the other hand, apply rate control by using the buffer status and achieve better results in error-prone environments. The choice of motion vector is incorporated within the rate-distortion in [170]. Another recent work about Rate-Distortion optimization for JVT/H.26L Video Coding in packet

	QP first	QP remain.	i.-GOB period	i.-frame period	Bitrate [kbit/s]
F.	17	17			158.21
F. i.-GOB	17	20(even)/21(odd)	1		139.31
F. i.-frames	17	23		9	140.82
A.	21	21			18.68
A. i.-GOB	21	25	4		22.21
A. i.-frames	21	26		36	22.34

Table 3.2: MSVC+intra-updates, Coding Parameters

loss environments is given in [132]. A unified rate-distortion analysis framework for transform coding is introduced in [69].

The following figures show the frame PSNR's over frame number for three error resilience techniques: 1- MSVC, 2- MSVC + GOB intra-updates and 3- MSVC + frame intra-updates for Foreman and Akiyo sequences at diverse loss rates. The coding parameters for both sequences and three techniques are given in Table 3.2. Both types of updates are performed periodically: 1 intra-coded GOB at every frame for Foreman and at every fourth frame for Akiyo, similarly 1 intra-coded frame at every 9^{th} frame for Foreman and at every 36^{th} frame for Akiyo. Each time, starting from the first GOB line of the frame, the next GOB position downwards is processed and intra-updates are inserted for both of the streams. Since Akiyo has a high prediction gain and a small coding rate, intra coding period is larger. The intra-coded frame of the odd sequence comes directly after the intra-coded frame of the even sequence. Increasing the temporal distance between the corresponding intra-updates of the two streams while keeping the same intra-period would certainly enhance the state recovery property, but this point is not investigated here.

The average frame PSNR, $PSNR_{avg}$ for every coding option and sequence and different loss rate combinations (p_1, p_2) is listed in Tables 3.3 and 3.4.

Figures 3.45 and 3.46 show the comparison of the 3 coding options for Foreman and Akiyo at the lossless case. The intra coded frames are coded at every coding option with the same quantization stepsize (QP): 17 for Foreman and 21 for Akiyo. The first option has the highest average PSNR, since it codes everything in inter-mode. The third option has the highest variation of frame PSNR since the QP's of odd and even sequence differ slightly (20 vs. 21). For Akiyo, the option with GOB-intra-updates has the smallest average PSNR. Figures 3.47 and 3.48 depict

	0%,0%	**0%,5%**	**0%,10%**	**0%,20%**	**5%,5%**
F.	35.19	34.45	34.28	34.17	32.23
F. i.-GOB	32.96	32.69	32.57	32.50	31.74
F. i.-frames	33.59	33.45	33.33	33.17	33.18
A.	35.00	34.81	34.81	34.84	34.19
A. i.-GOB	32.62	32.52	32.51	32.53	32.28
A. i.-frames	34.03	33.96	33.93	33.94	33.80

Table 3.3: MSVC, $PSNR_{avg}$ depending on (p_1, p_2)

	5%,10%	**5%,20%**	**10%,10%**	**10%,20%**	**20%,20%**
F.	30.41	29.64	28.79	27.33	24.93
F. i.-GOB	31.02	30.65	30.04	29.26	27.26
F. i.-frames	32.85	32.55	32.48	31.91	30.84
A.	33.90	33.77	33.45	33.15	32.43
A. i.-GOB	32.15	32.09	31.95	31.82	31.43
A. i.-frames	33.68	33.61	33.48	33.33	32.89

Table 3.4: MSVC, $PSNR_{avg}$ depending on (p_1, p_2)

the case when the first stream is received lossless and the second one is lossy with $p_2 = 0.05$. We see that flickers occur between reconstructed odd and even sequences, but state recovery works. The flickers are smaller for Akiyo, due to the success of error concealment. When p_2 increases while p_1 kept constant, the flickers increase and $PSNR_{avg}$ decreases as seen in Figures 3.49 and 3.51 for Foreman. Figures 3.50 and 3.52 for Akiyo depict that increasing p_2 does not cause any change as long as $p_1 = 0$.

But what happens when both of the streams are lossy? Figures 3.53 and 3.54 show the resulting frame PSNR when both stream are received with $p_1 = p_2 = 0.05$. It is seen that intra-updates supply more robustness for Foreman. By adjusting the bitrate trade-off between quantization step size and the intra-coding period, a better adaptation to channel conditions can be achieved. The GOB-intra-coding is not that efficient as the frame-intra-update coding. The resulting $PSNR_{avg}$ is lower than without intra-coding. For Akiyo, on the other hand, using intra-coding is disadvantageous, since the available bitrate is low and the interpolation works well.

When we increase p_2 from 0.05 to 0.1 or to 0.2 as seen in Figures 3.55 and in 3.57, the importance of intra-coding for Foreman increases, even the GOB-intra-updates

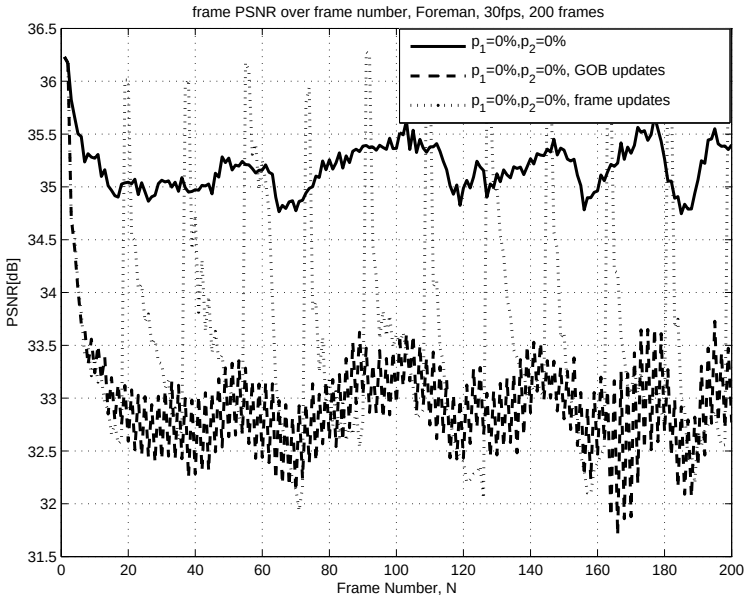


Figure 3.45: PSNR over Frame Number, Approach 1, Foreman, interleaved, lossless

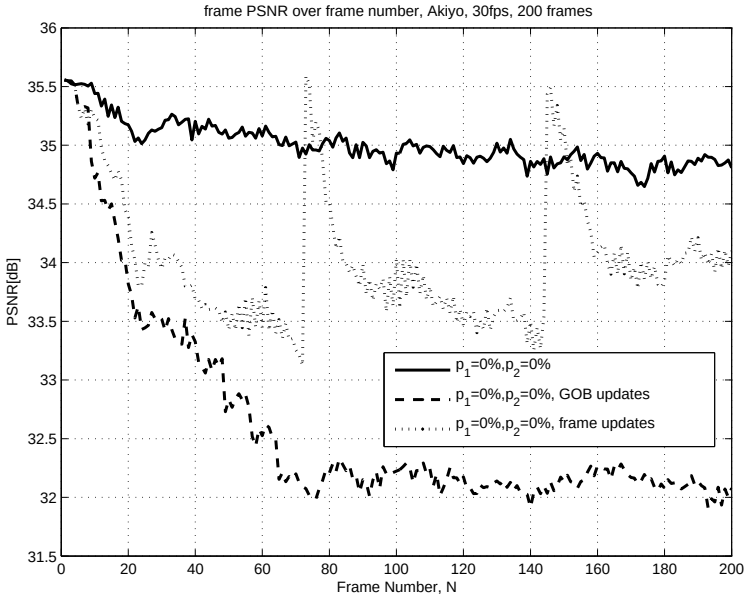


Figure 3.46: PSNR over Frame Number, Approach 1, Akiyo, interleaved, lossless

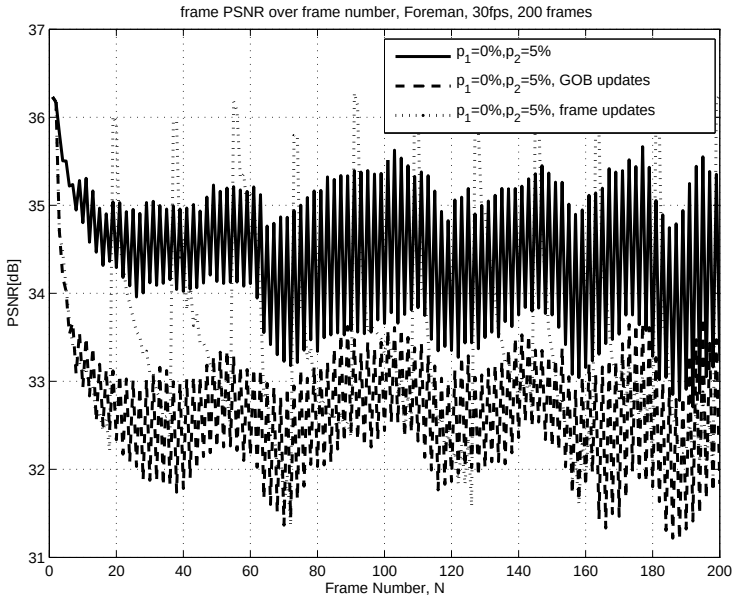


Figure 3.47: PSNR over Frame Number, Approach 1, Foreman, interleaved, first stream lossless

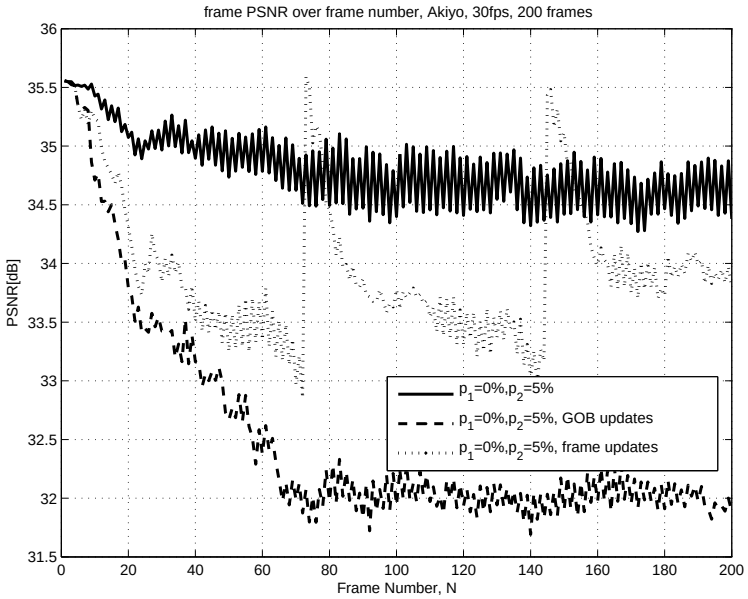


Figure 3.48: PSNR over Frame Number, Approach 1, Akiyo, interleaved, first stream lossless

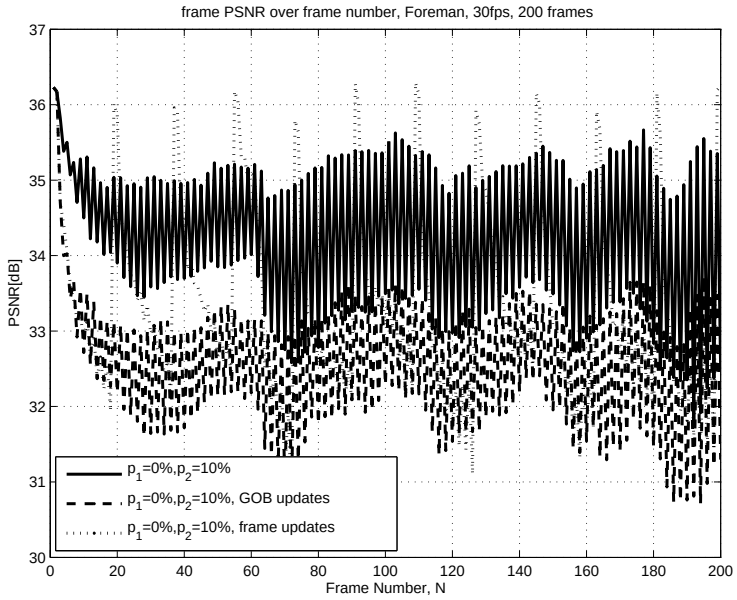


Figure 3.49: PSNR over Frame Number, Approach 1, Foreman, interleaved, first stream lossless

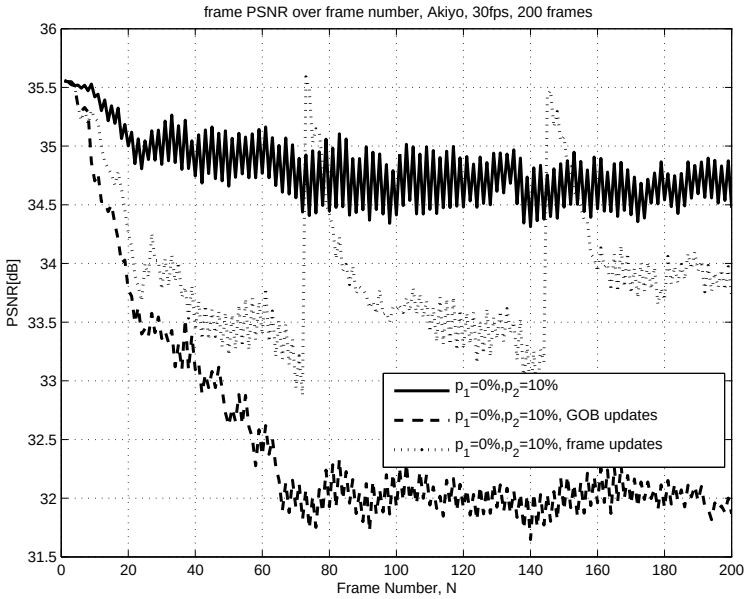


Figure 3.50: PSNR over Frame Number, Approach 1, Akiyo, interleaved, first stream lossless

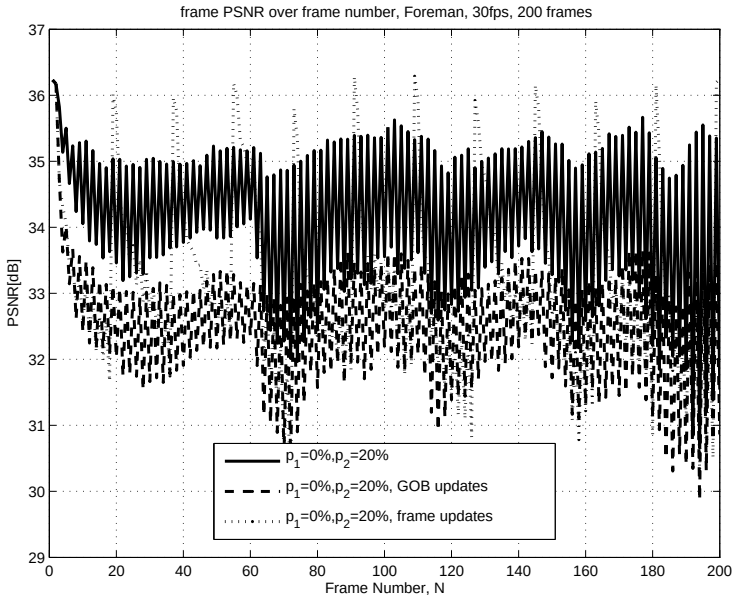


Figure 3.51: PSNR over Frame Number, Approach 1, Foreman, interleaved, first stream lossless

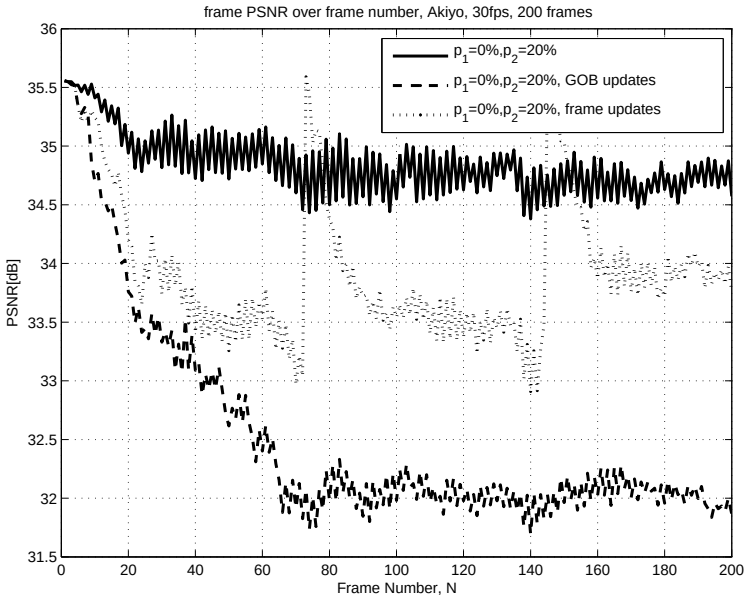


Figure 3.52: PSNR over Frame Number, Approach 1, Akiyo, interleaved, first stream lossless

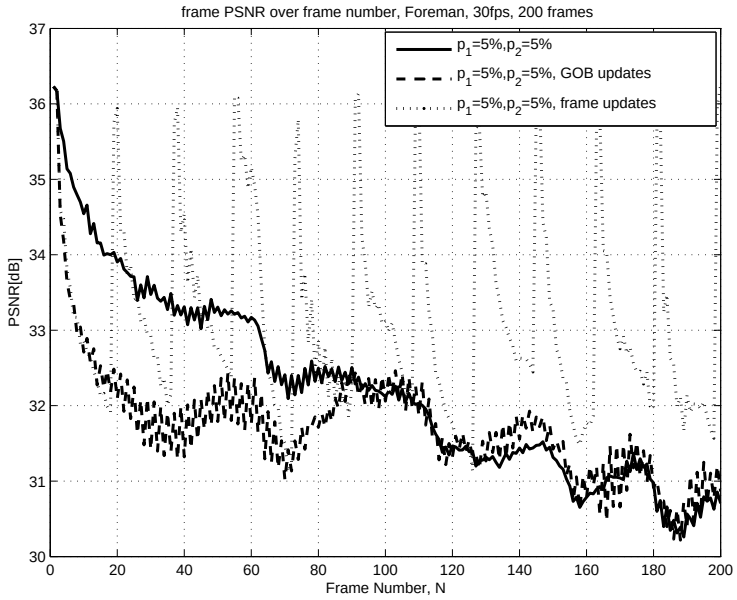


Figure 3.53: PSNR over Frame Number, Approach 1, Foreman, interleaved, lossy

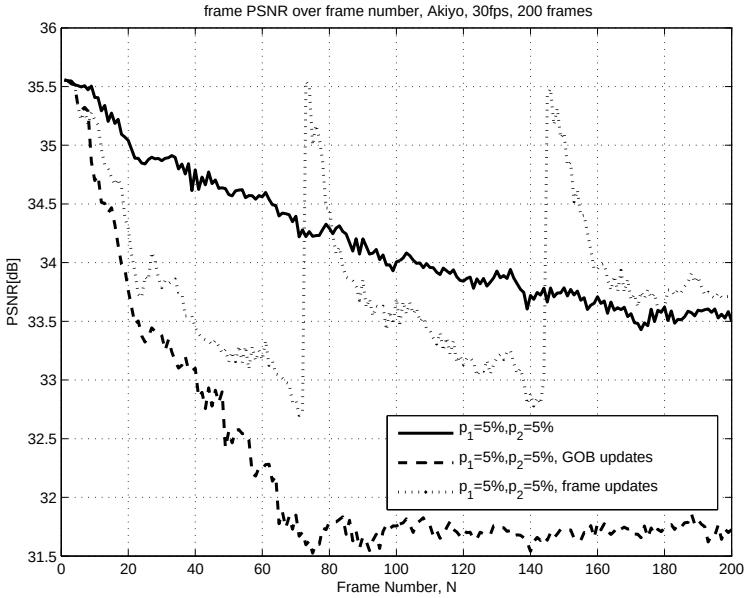


Figure 3.54: PSNR over Frame Number, Approach 1, Akiyo, interleaved, lossy

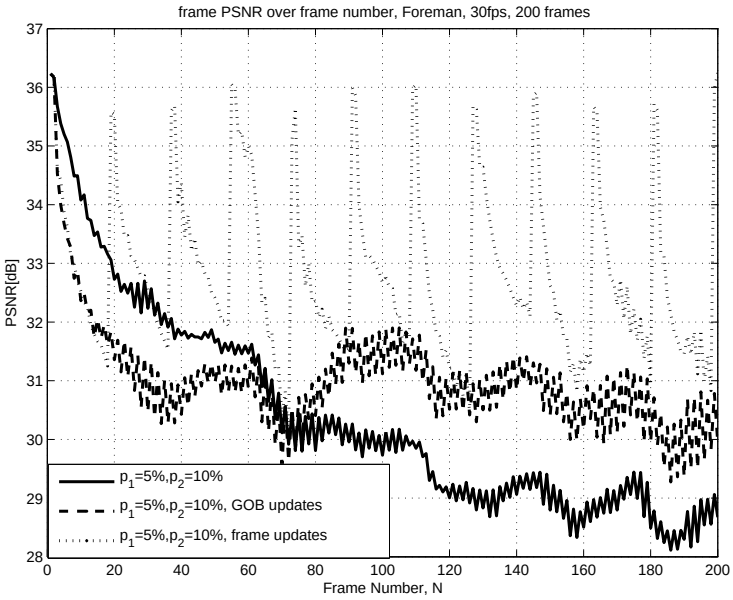


Figure 3.55: PSNR over Frame Number, Approach 1, Foreman, interleaved, lossy

case becomes advantageous as the frame number increases. For Akiyo, the intra-coding at the current setting does not really pays off as long as $p_1 = 0.05$. Non-intra coding combined with state recovery yields acceptable results, Figures 3.56 and 3.58 confirm this.

Figures 3.59 and 3.60 show the case when $p_1 = 0.1$. Intra-frame updates yield the best results for Foreman followed by the GOB-intra-updates. For Akiyo however, no-intra-updates are still better than GOB-intra-updates. The same effect is observed when loss rates p_1 and p_2 increase from 0.1 to 0.2. The frame PSNR values are summarized in Tables 3.3 and 3.4.

3.4.4 Unbalanced Quantized MSVC

Until now, we investigated the case that both streams are coded at the same bitrate, at the same frame rate and also at the same quantization stepsize, giving the same $PSNR_{avg}$ for both streams. But sometimes it could make sense to allocate more bitrate to the more promising path in terms of probable losses. In another scenario, one of the paths could have more bandwidth available at a certain point. Unbalanced video coding in context of MDC is not widely explored. One of the papers investigating unbalanced MD video coding is [21] where the system produces two

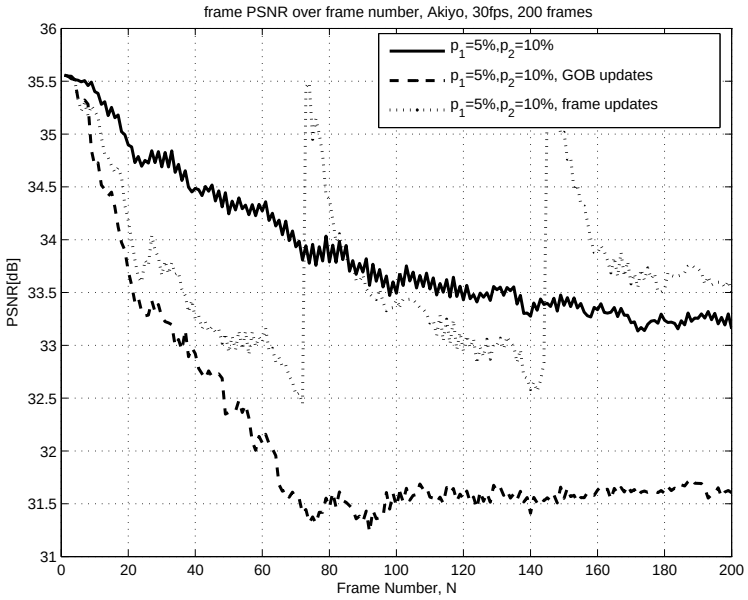


Figure 3.56: PSNR over Frame Number, Approach 1, Akiyo, interleaved, lossy

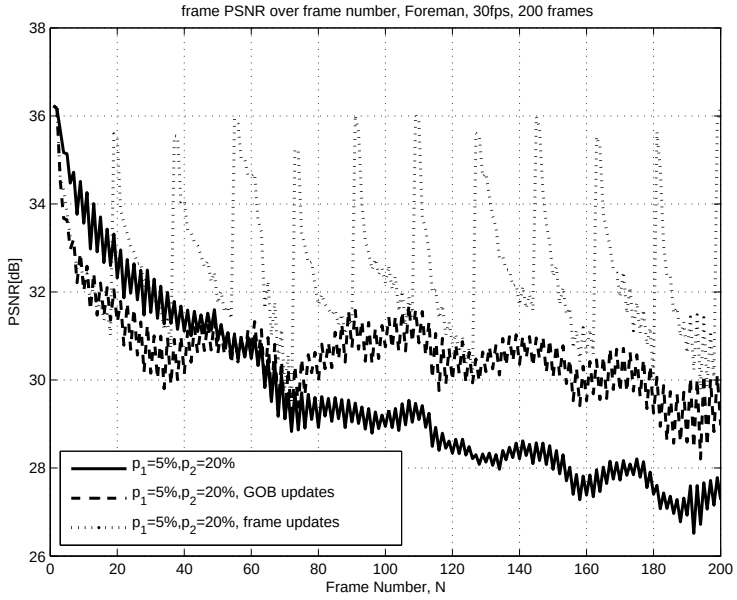


Figure 3.57: PSNR over Frame Number, Approach 1, Foreman, interleaved, lossy

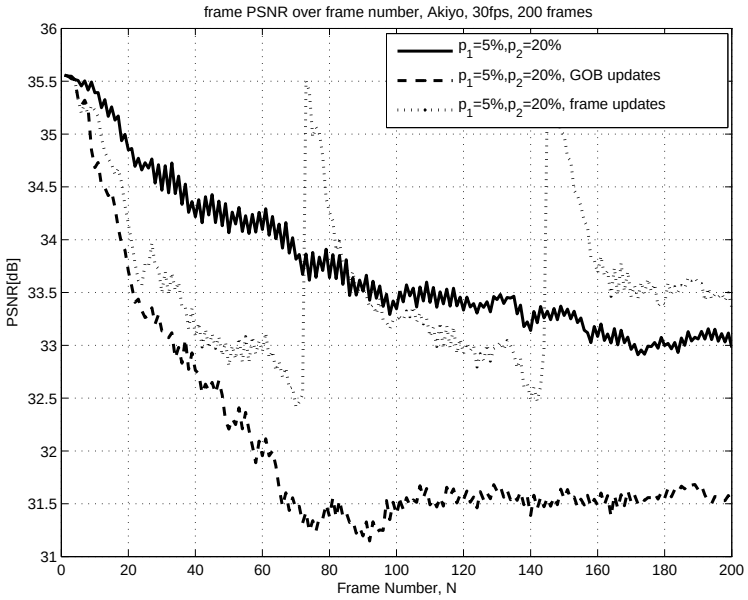


Figure 3.58: PSNR over Frame Number, Approach 1, Akiyo, interleaved, lossy

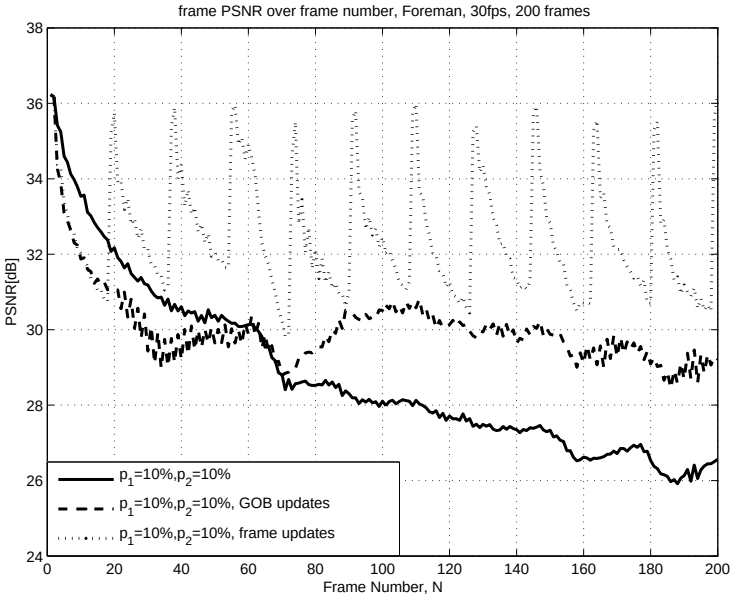


Figure 3.59: PSNR over Frame Number, Approach 1, Foreman, interleaved, lossy

Figure 3.60: PSNR over Frame Number, Approach 1, Akiyo, interleaved, lossy

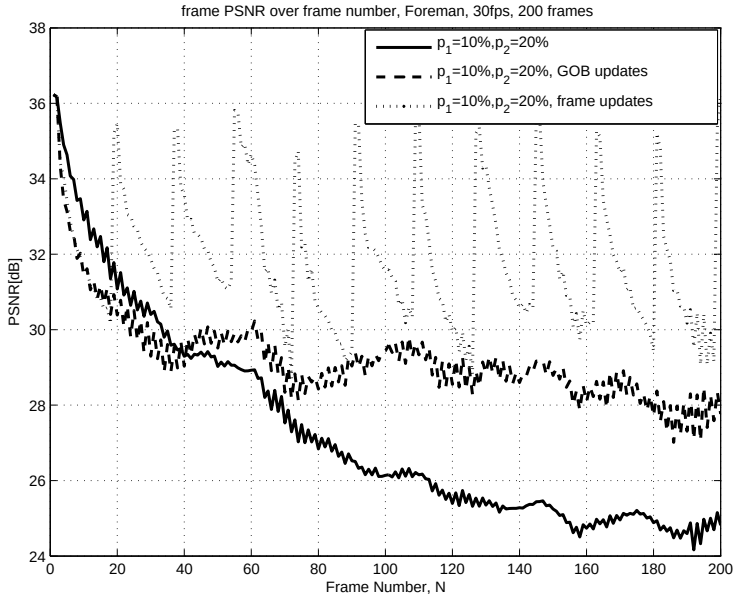


Figure 3.61: PSNR over Frame Number, Approach 1, Foreman, interleaved, lossy

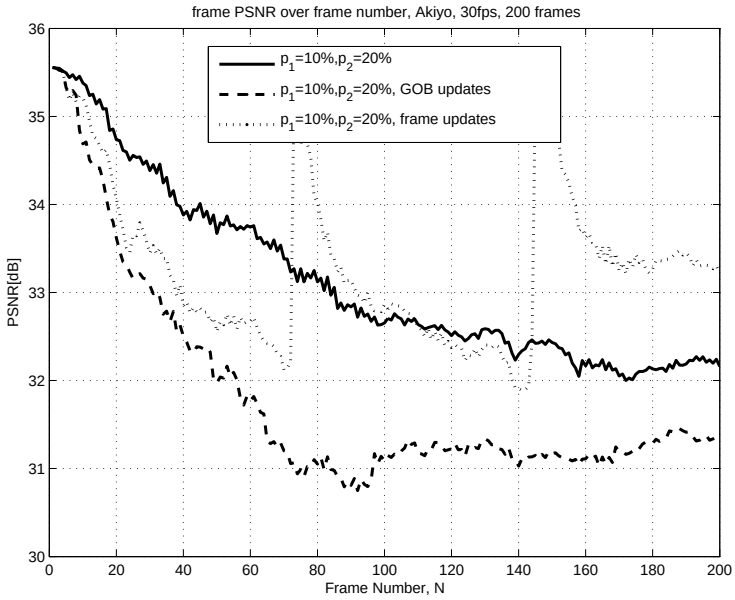


Figure 3.62: PSNR over Frame Number, Approach 1, Akiyo, interleaved, lossy

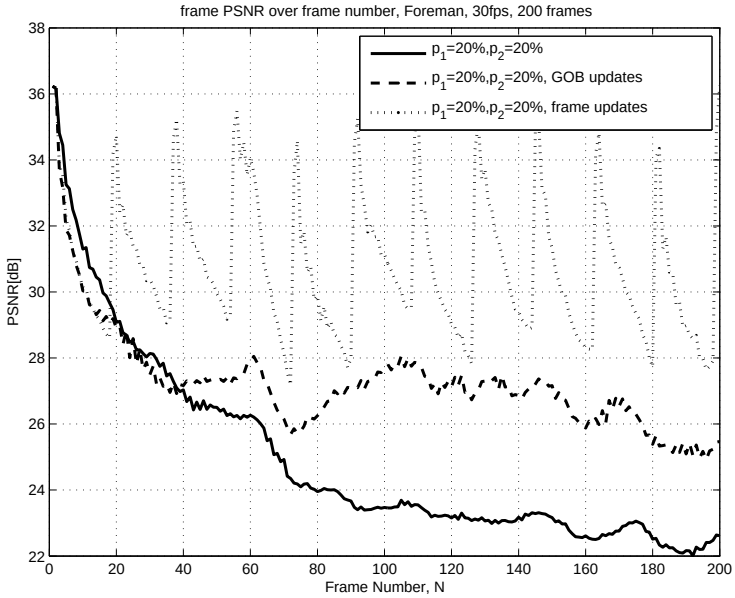


Figure 3.63: PSNR over Frame Number, Approach 1, Foreman, interleaved, lossy

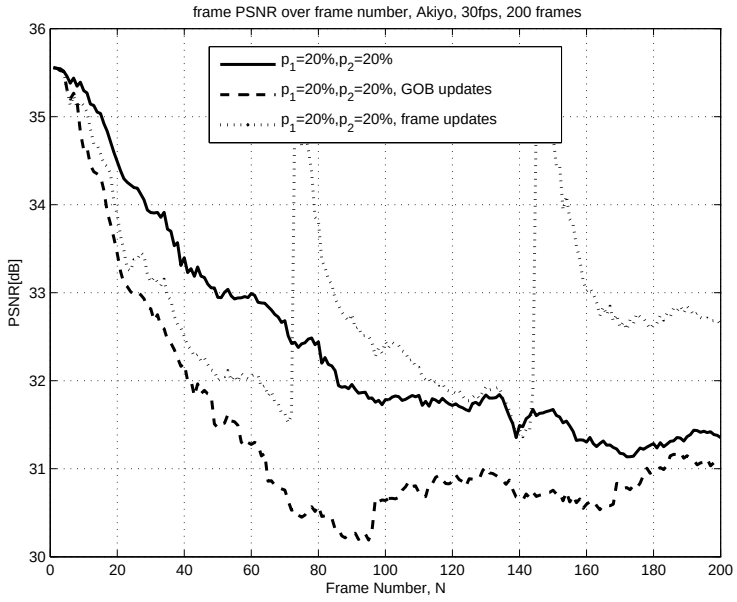


Figure 3.64: PSNR over Frame Number, Approach 1, Akiyo, interleaved, lossy

descriptions with different resolutions of transform coefficients. The lower resolution description is not meant to be usable by itself and is only used to replace the lost data in the high resolution description. By its description, the work described in this paper corresponds to layered coding rather than MDC, since MDC requires independent decoding of all its descriptions.

Another work on unbalanced video coding is given in [5] where MSVC is extended to an unbalanced system based on frame rate adaptation. The motivation is network paths with unbalanced bandwidths demanding unbalanced bitrates for the transmission. The video is coded into two streams producing unbalanced frame rates of 2:1.

Other methods one might think of to generate unbalanced descriptions are based on adaptation of the quantization or the spatial resolution of the framewise splitted video signal. However approximately equal quality streams are important for visual perception so that a quality variation (flicker) at half the original frame rate is avoided, especially if there are no losses at all. In [5] it is mentioned that rate adaptation via coarser quantization is inappropriate for large rate changes due to flickers and may only be used for small rate changes (PSNR differences of up to 0.5 dB between streams). It is argumented that frame rate adaptation is a more effective mechanism for unbalanced MDC since the frame rate variations are visually less disturbing than

	QP_1	$R_1[kbits/s]$	QP_2	$R_2[kbits/s]$
Coding Option 1	14	111.88	26	27.48
Coding Option 2	15	95.94	21	43.01
Coding Option 3	16	83.16	19	54.91
Coding Option 4	17	71.38	17	71.04

Table 3.5: Unbalanced Quantized MSVC, Coding Parameters, Foreman

	QP_1	$R_1[kbits/s]$	QP_2	$R_2[kbits/s]$
Coding Option 1	17	106.38	27	34.02
Coding Option 2	18	94.55	24	46.98
Coding Option 3	19	83.27	22	59.14
Coding Option 4	20	73.83	21	65.68

Table 3.6: Unbalanced Quantized MSVC, Coding Parameters, Foreman with GOB-intra-updates

quality variations. Here we will investigate the unbalanced MSVC based on quantization adaptation presented also in [34]. Figures 3.65, 3.66 and 3.67 shows the case when the first stream is received lossless and the second one is lossy with loss rates of 5%, 10% and 20%. The bitrate allocated to the first stream is varied along the x-axis from half the total bitrate R_T to nearly the full bitrate R_T. R_T is set to 140 kbit/s in the experiment. The bitrate allocation is unbalanced via changing the quantization stepsizes QP_1 and QP_2. Tables 3.5 3.6, 3.7, 3.8, 3.9 and 3.10 list the QP's chosen for coding the sequences Foreman and Akiyo without and with intra-coding of GOB's and frames. The intra-coding periods are given in Table 3.2.

Figure 3.65 shows the $PSNR_{avg}$ variation for coding without intra-coding whereas 3.66 and 3.67 for intra-updates with GOB's and frames respectively. We observe that as p_2 increases $R_{1,opt}$, rate yielding the maximum average frame PSNR after interleaving, increases, i.e. unbalance increases. The shift in unbalance is larger if no

	QP_1	$R_1[kbits/s]$	QP_2	$R_2[kbits/s]$
Coding Option 1	20	83.90	29	55.68
Coding Option 2	21	78.23	26	61.36
Coding Option 3	22	74.04	24	66.73
Coding Option 4	23	70.24	23	70.24

Table 3.7: Unbalanced Quantized MSVC, Coding Parameters, Foreman with frame-intra-updates

	QP_1	$R_1[kbits/s]$	QP_2	$R_2[kbits/s]$
Coding Option 1	18	13.92	27	5.03
Coding Option 2	19	12.16	24	6.75
Coding Option 3	20	10.64	22	8.41
Coding Option 4	21	9.41	21	9.41

Table 3.8: Unbalanced Quantized MSVC, Coding Parameters, Akiyo

	QP_1	$R_1[kbits/s]$	QP_2	$R_2[kbits/s]$
Coding Option 1	22	13.22	30	5.97
Coding Option 2	23	11.67	28	7.14
Coding Option 3	24	10.59	26	8.72
Coding Option 4	25	9.64	25	9.64

Table 3.9: Unbalanced Quantized MSVC, Coding Parameters, Akiyo with GOB-intra-updates

	QP_1	$R_1[kbits/s]$	QP_2	$R_2[kbits/s]$
Coding Option 1	23	11.26	30	7.97
Coding Option 2	24	10.56	28	8.61
Coding Option 3	25	9.95	27	8.97
Coding Option 4	26	9.45	26	9.45

Table 3.10: Unbalanced Quantized MSVC, Coding Parameters, Akiyo with frame-intra-updates

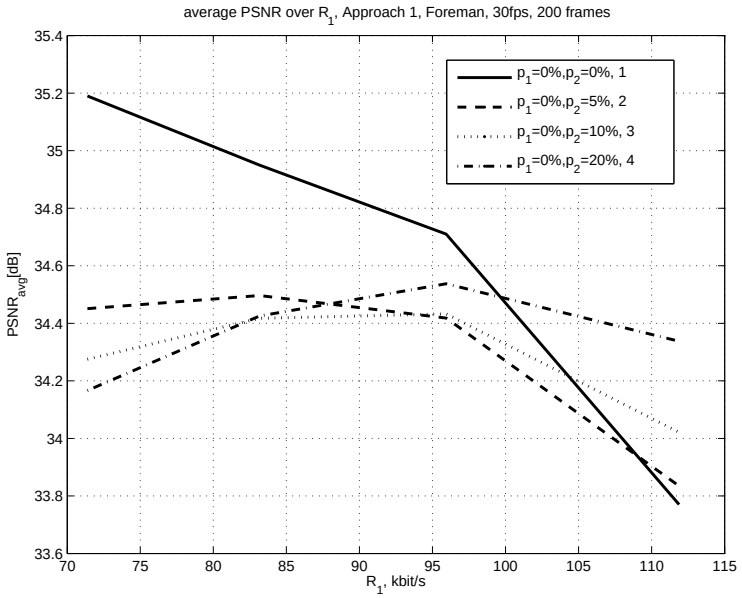


Figure 3.65: $PSNR_{avg}$ over R_1, Approach 1, Unbalanced, Foreman, first stream lossless

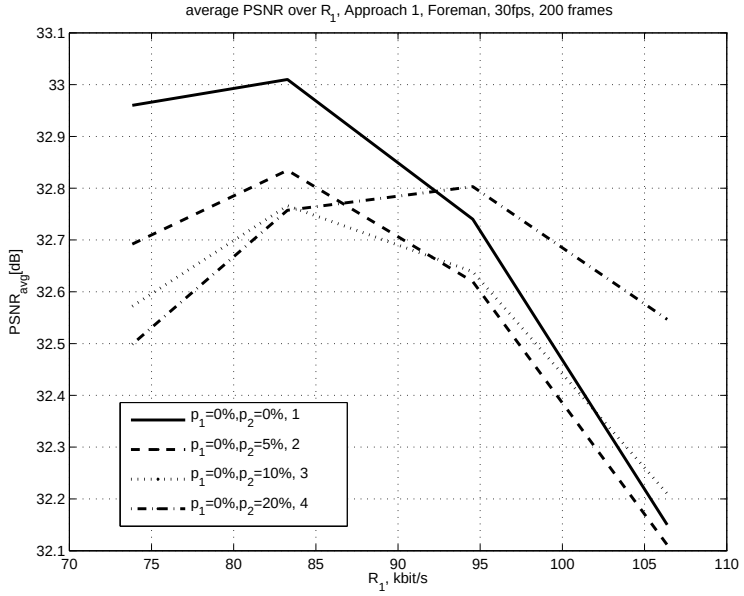


Figure 3.66: $PSNR_{avg}$ over R_1, Approach 1, Unbalanced, Foreman with GOB-intra-updates, first stream lossless

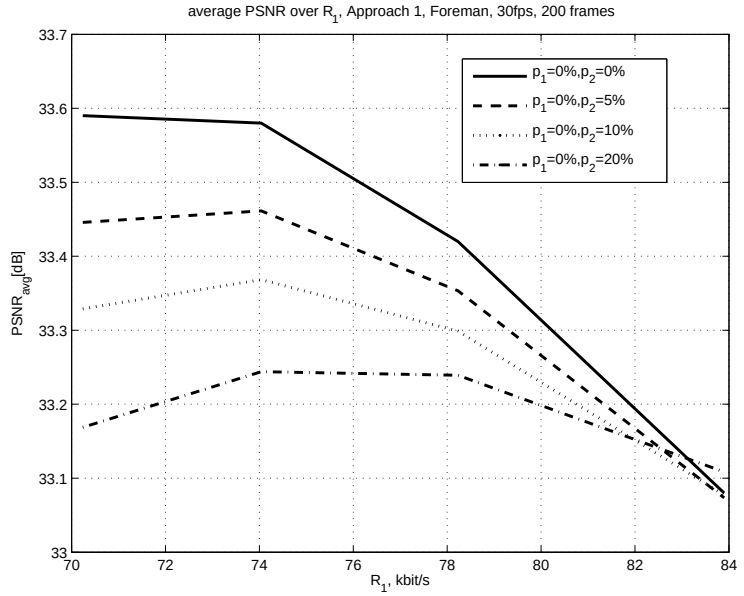


Figure 3.67: $PSNR_{avg}$ over R_1, Approach 1, Unbalanced, Foreman with frame-intra-updates, first stream lossless

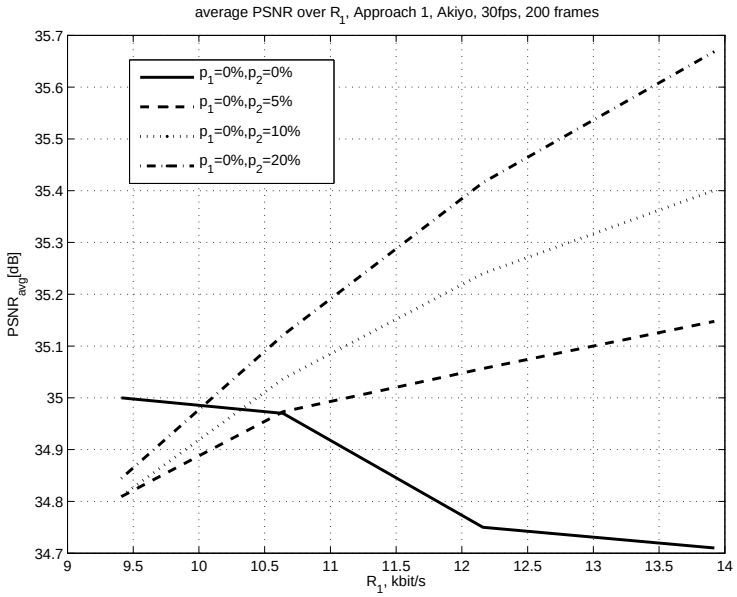


Figure 3.68: $PSNR_{avg}$ over R_1, Approach 1, Unbalanced, Akiyo, first stream lossless

intra-coding is used and smallest if frame-intra-updates are employed.

Figures 3.68, 3.69 and 3.70 show the result of the same experiment for Akiyo: The observation is the same as for Foreman. But one additional point becomes more obvious: as p_2 increases $PSNR_{avg}$ also increases if bitrate allocation is unbalanced. This effect is easily explained: Instead of using a low rate stream it could be advantageous to discard it and use interpolations from the higher rate stream (comparison of $(0\%, 0\%)$ with $(0\%, 20\%)$ as R_1 increases). This observation leads us from Approach 1 to Approach 2 which will be discussed in the next subsection.

Figures 3.71, 3.72 and 3.73 show $PSNR_{avg}$ over R_1 for balanced loss probabilities for Foreman and 3.74, 3.75 and 3.76 the same for Akiyo.

The performance of Approach 1 at unbalanced loss probabilities is demonstrated in Figures 3.77, 3.78 and 3.79 for Foreman and in 3.80, 3.81 and 3.82 for Akiyo. For Foreman, excluding frame-intra-updates, the optimal operating point coincides with balanced rate allocations. When frame-intra-updates are used, the length of the sequence coded without intras is small so that the high rate stream can help the low rate stream to recover from losses. But as the sequence length increases the interaction between streams increases and we need additionally the low rate stream to help the high rate stream to recover, i.e. the unbalance becomes disadvantageous.

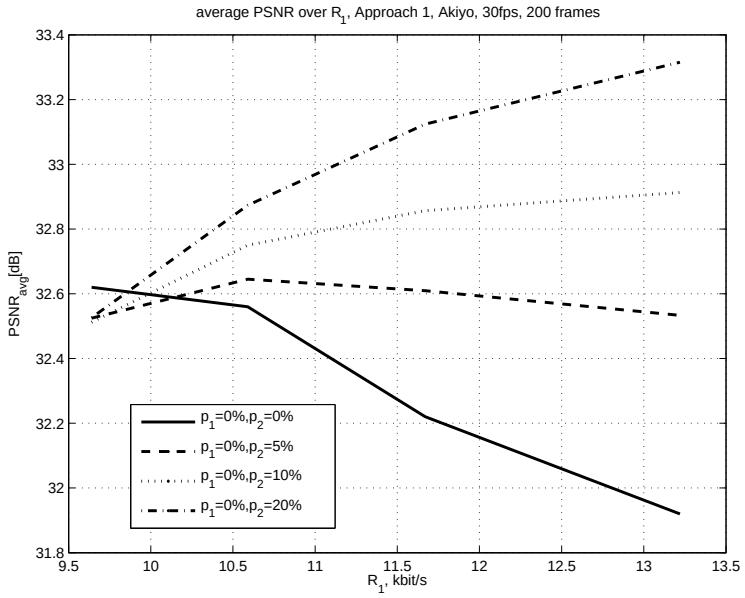


Figure 3.69: $PSNR_{avg}$ over R_1, Approach 1, Unbalanced, Akiyo with GOB-intra-updates, first stream lossless

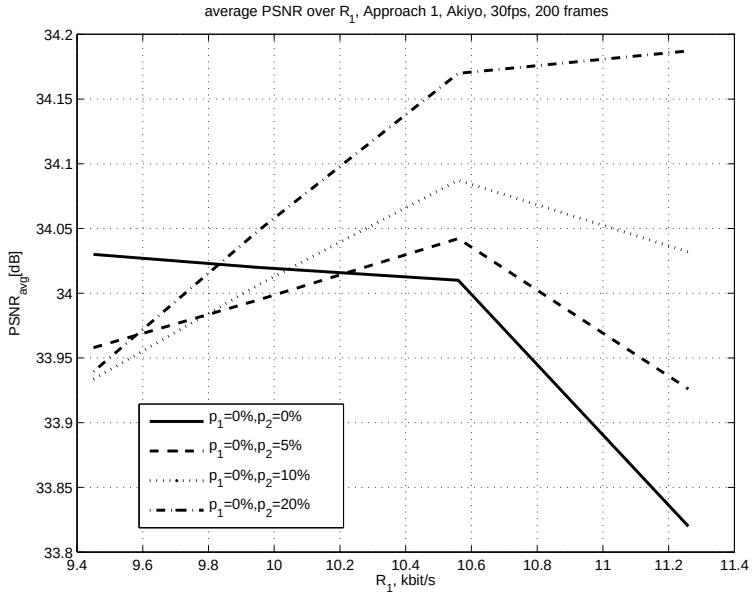


Figure 3.70: $PSNR_{avg}$ over R_1, Approach 1, Unbalanced, Akiyo with frame-intra-updates, first stream lossless

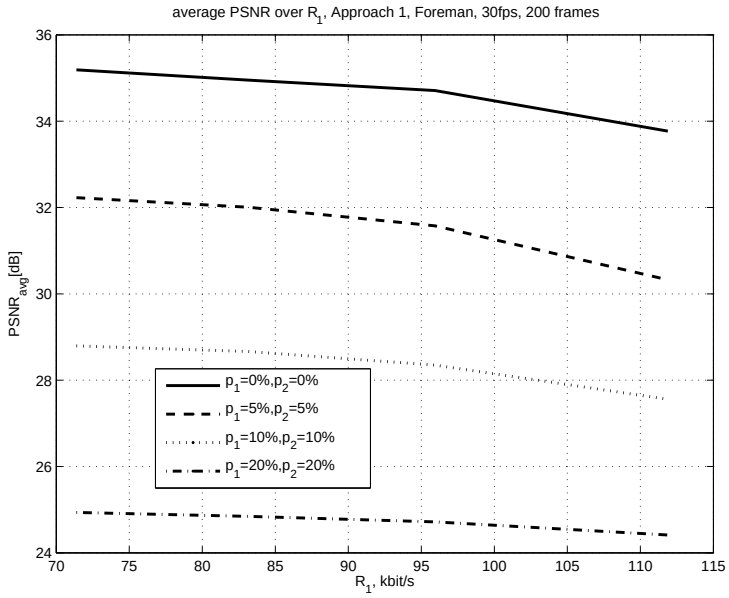


Figure 3.71: $PSNR_{avg}$ over R_1, Approach 1, Unbalanced, Foreman, balanced loss probabilities

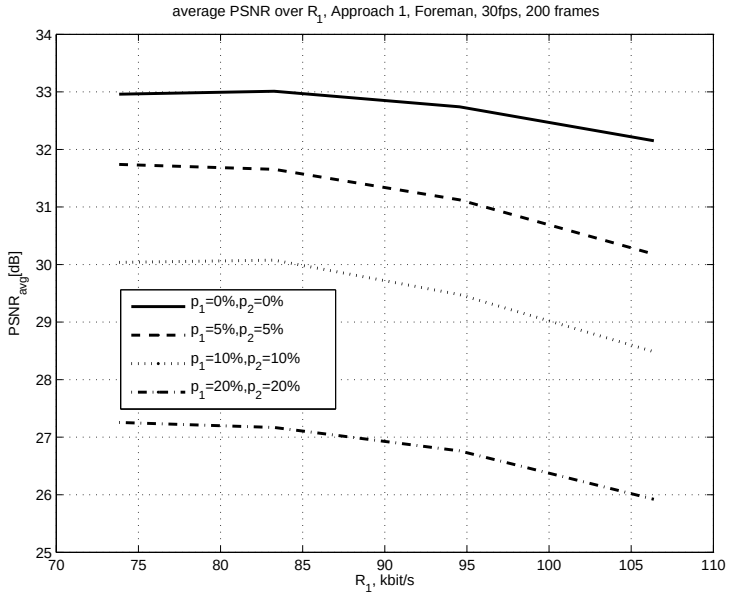


Figure 3.72: $PSNR_{avg}$ over R_1, Approach 1, Unbalanced, Foreman with GOB-intra-updates, balanced loss probabilities

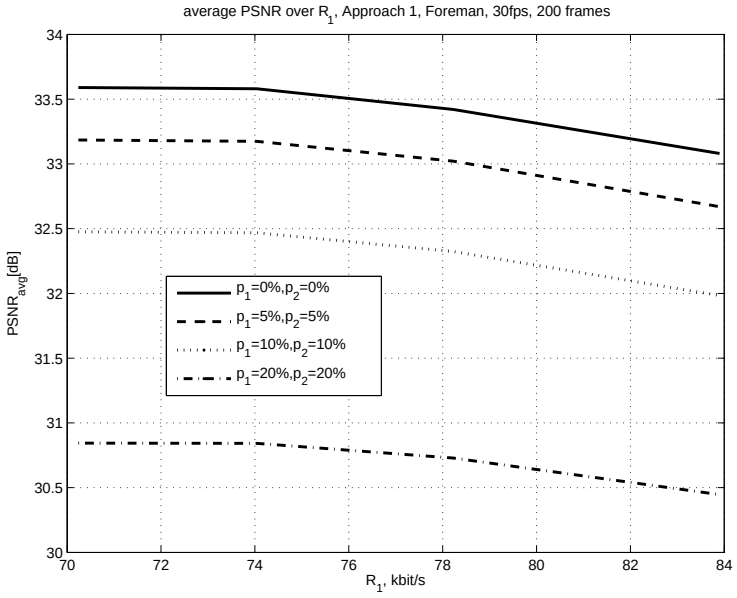


Figure 3.73: $PSNR_{avg}$ over R_1, Approach 1, Unbalanced, Foreman with frame-intra-updates, balanced loss probabilities

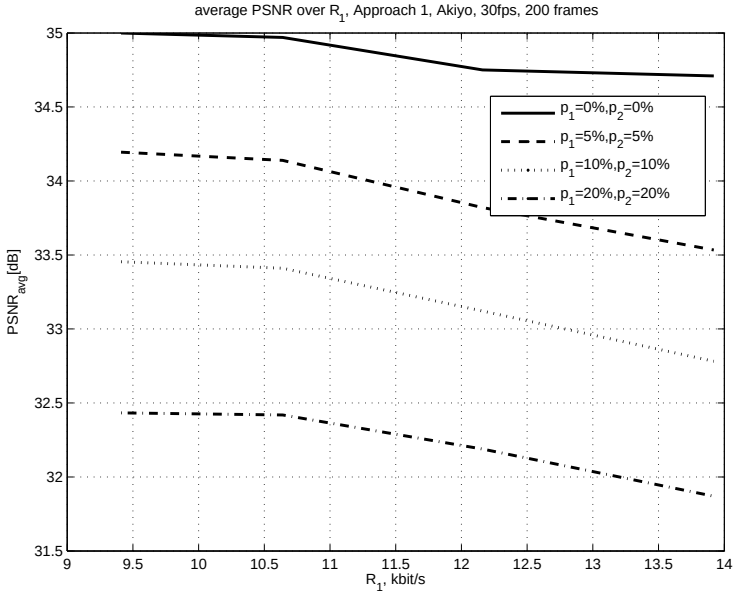


Figure 3.74: $PSNR_{avg}$ over R_1, Approach 1, Unbalanced, Akiyo, balanced loss probabilities

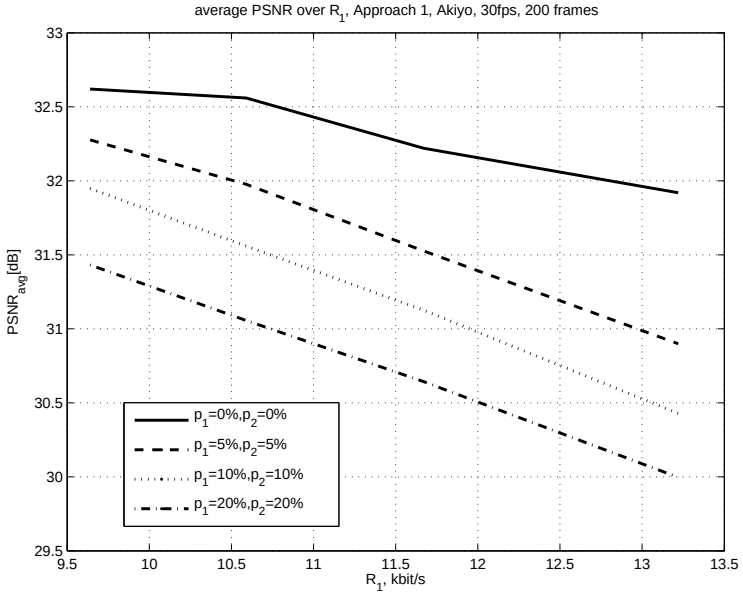


Figure 3.75: $PSNR_{avg}$ over R_1, Approach 1, Unbalanced, Akiyo with GOB-intra-updates, balanced loss probabilities

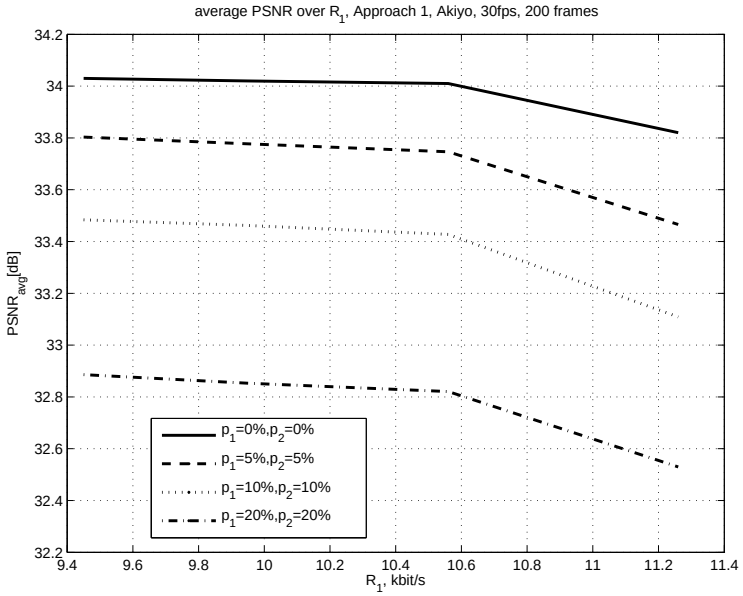


Figure 3.76: $PSNR_{avg}$ over R_1, Approach 1, Unbalanced, Akiyo with frame-intra-updates, balanced loss probabilities

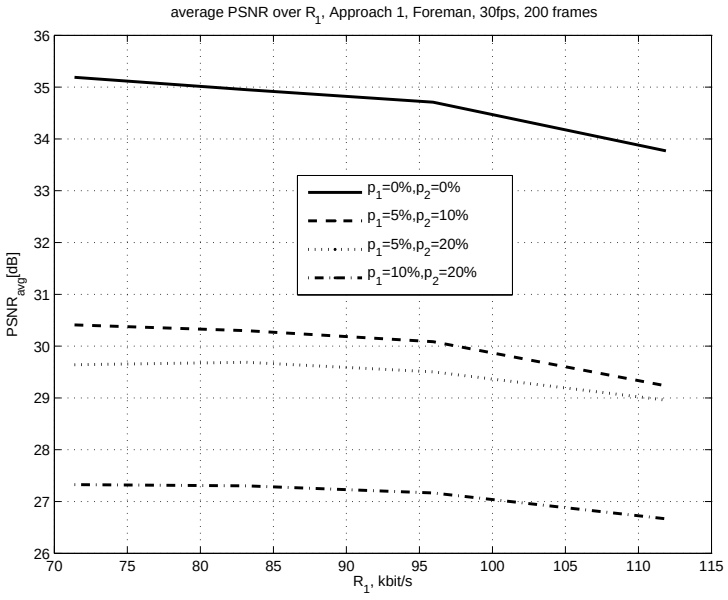


Figure 3.77: $PSNR_{avg}$ over R_1, Approach 1, Unbalanced, Foreman, unbalanced loss probabilities

The same observation holds also for Akiyo. Unbalancing the system pays off only when intra-frame-coding is used, i.e. sequence length until the next state refreshment is small.

3.5 Approach 2

In the previous subsection, we have seen when $p_1 = 0$, increasing p_2 surprisingly increases $PSNR_{avg}$. This can be explained by the fact that the error due to error propagation is sometimes larger than the error due to frame interpolation. Approach 2 extends Approach 1 in the sense that when a packet corresponding to a frame is received, we have two options: 1- use the data in the packet for the reconstruction, 2- discard it and reconstruct the frame by interpolating its previous and next neighbors from the other thread. Certainly we can gain even better results when we can use the data in the packet to enhance the interpolation process, but this aspect will be explored in the future work. One method to enhance the quality of interpolated images using coarsely quantized images as side information is discussed in [1] in context of Wyner-Ziv Coding and conditionally interdecoding of individually encoded

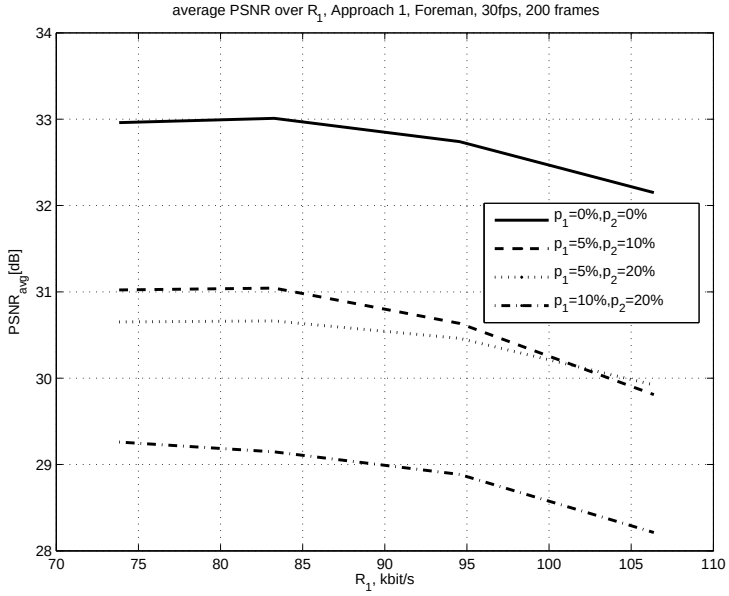


Figure 3.78: $PSNR_{avg}$ over R_1, Approach 1, Unbalanced, Foreman with GOB-intra-updates, unbalanced loss probabilities

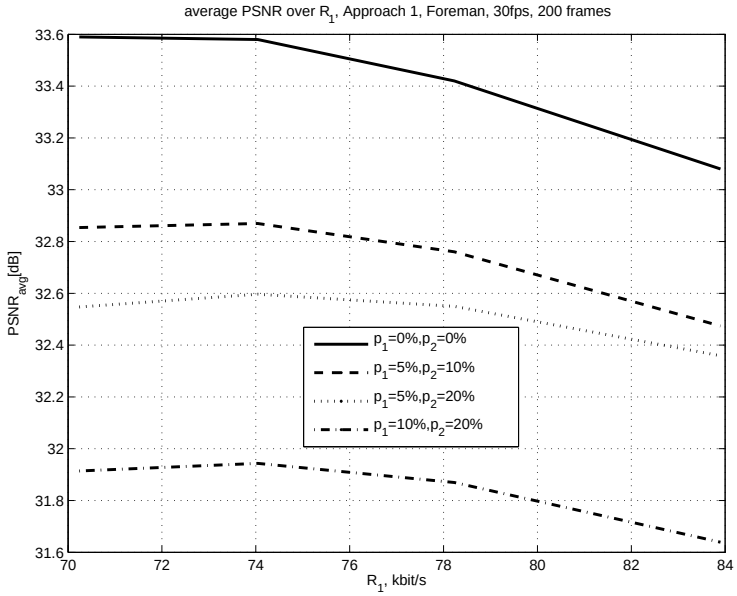


Figure 3.79: $PSNR_{avg}$ over R_1, Approach 1, Unbalanced, Foreman with frame-intra-updates, unbalanced loss probabilities

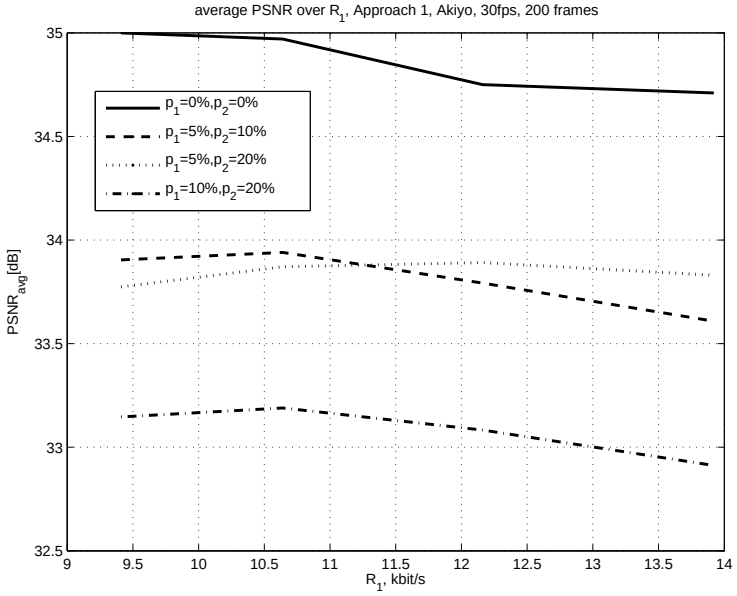


Figure 3.80: $PSNR_{avg}$ over R_1, Approach 1, Unbalanced, Akiyo, unbalanced loss probabilities

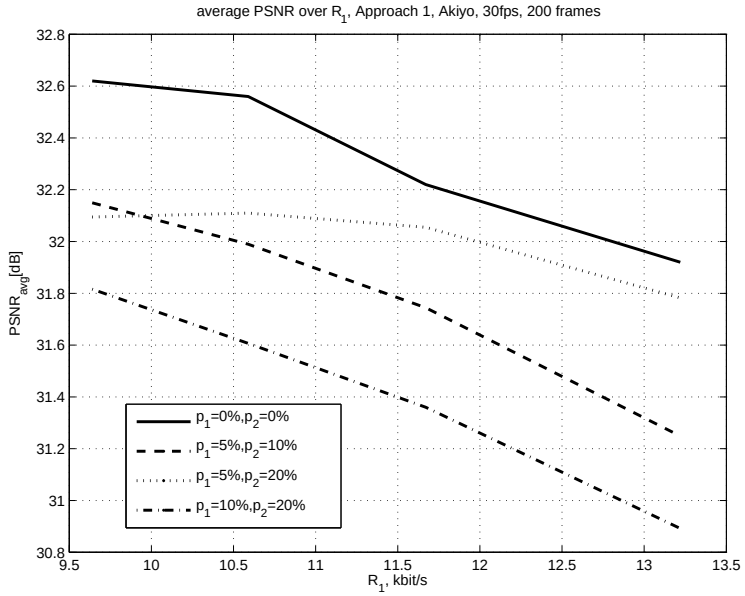


Figure 3.81: $PSNR_{avg}$ over R_1, Approach 1, Unbalanced, Akiyo with GOB-intra-updates, unbalanced loss probabilities

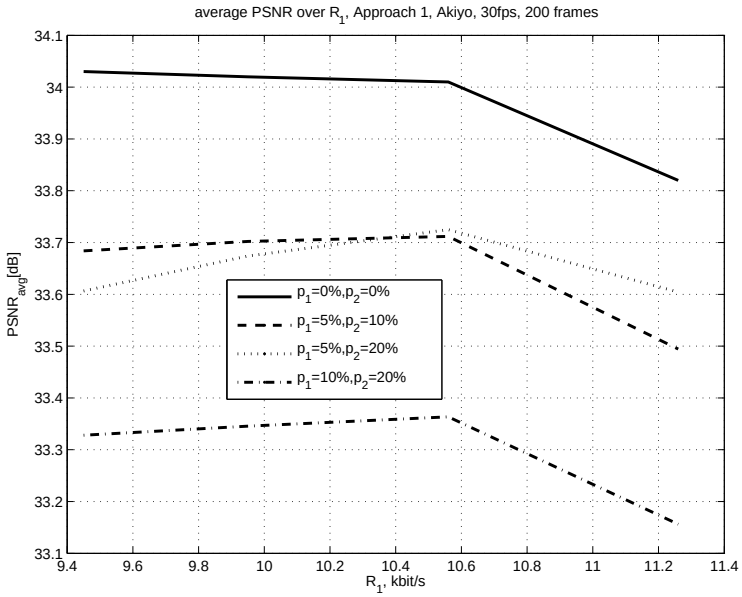


Figure 3.82: $PSNR_{avg}$ over R_1, Approach 1, Unbalanced, Akiyo with frame-intra-updates, unbalanced loss probabilities

(intra-encoded) frames. A similar approach can be adapted also here for intercoded frames and their conditional decoding to further improve the reconstruction results.

Approach 2, as described in the previous paragraph is clearly suboptimal. We look only one step ahead and try to select among two options for frames whose corresponding packets are received: use the packet or use the interpolation result. If we try to go N steps ahead and try to choose the optimum steps to maximize $PSNR_{avg}$ we would have a tree structure of depth N. However increasing the depth increases the complexity. Therefore we have chosen depth of 1 for our further exploration.

Figures 3.83 and 3.84 summarize the motivation for Approach 2. Here $PSNR_{avg}$ over R_1 is plotted for two cases: 1-Approach 1, 2-use always the interpolations, discard the second stream. We obtain larger $PSNR_{avg}$ by allocating more bitrate to the first stream and interpolating the second thread using the data from the first one, rather than by halving the bitrate equally and applying Approach 1. This motivates us not only towards unbalancing the resource allocation but also towards Approach 2. The figures show only the case that both streams are received lossless. The next question is what happens when the channels are lossy, what does Approach 2 bring in this case? Figures 3.85, 3.86 and 3.87 show for Foreman the RD performance of Approach

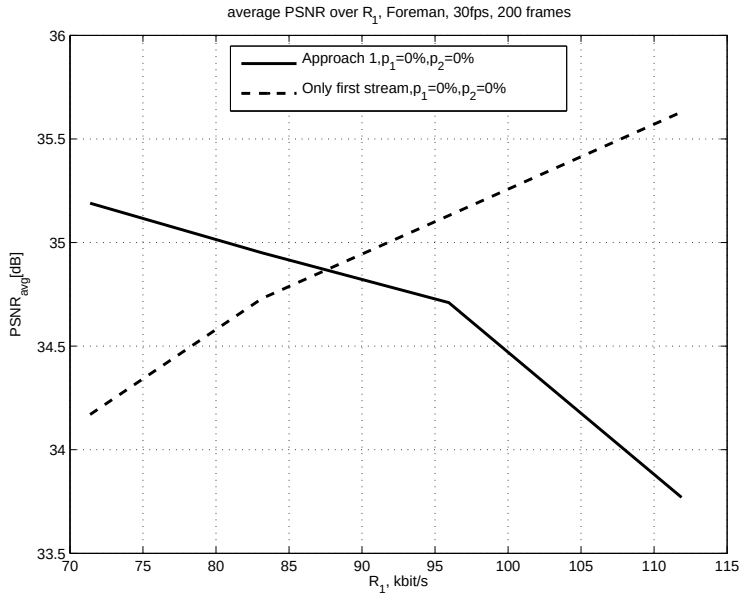


Figure 3.83: $PSNR_{avg}$ over R_1, Motivation for Approach 2, Foreman

2 for with and without intra-updates when the first stream is received lossless and the second one is lossy. We observe for no-intra-coding and lossy transmission, as the unbalance increases $PSNR_{avg}$ increases also. The maximum $PSNR_{avg}$ is achieved at balance (both streams coded at $70kbits/s$) only for the lossless transmission. With GOB-intra updates however, even at the lossless transmission, unbalance is advantageous. For frame-intra-updates, on the other hand, the maximum $PSNR_{avg}$ is shifted towards right only when p_2 increases beyond 5%.

Figures 3.88, 3.89 and 3.90 show the results of the same experiment for Akiyo. If coded without intras or with GOB-intra-updates independent of the loss rate, we get the same $PSNR_{avg}$ at every rate allocation. This is due to the successful error concealment. When using frame-intra-updates there are slight differences for different p_2 values. For both Foreman and Akiyo, bigger loss rates yield higher $PSNR_{avg}$ at high unbalance. The reason is that higher loss rates require/allow more interpolation than lower loss rates. Approach 2 in its current implementation is suboptimal as discussed before.

Next we investigate what happens when we have balanced loss probabilities. The results for Foreman and Akiyo with different coding options are demonstrated in figures 3.91, 3.92 and 3.93 and in 3.94, 3.95 and 3.96 respectively. We observe that in

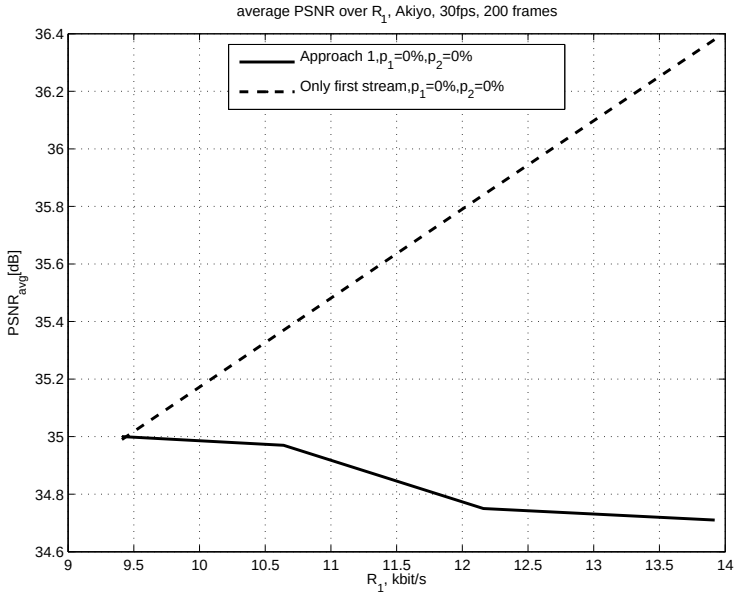


Figure 3.84: $PSNR_{avg}$ over R_1, Motivation for Approach 2, Akiyo

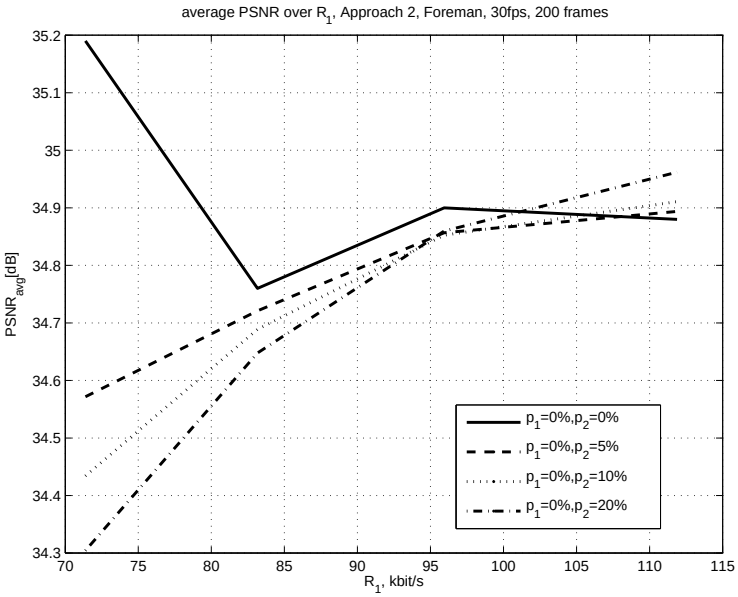


Figure 3.85: $PSNR_{avg}$ over R_1, Approach 2, Foreman, first stream lossless

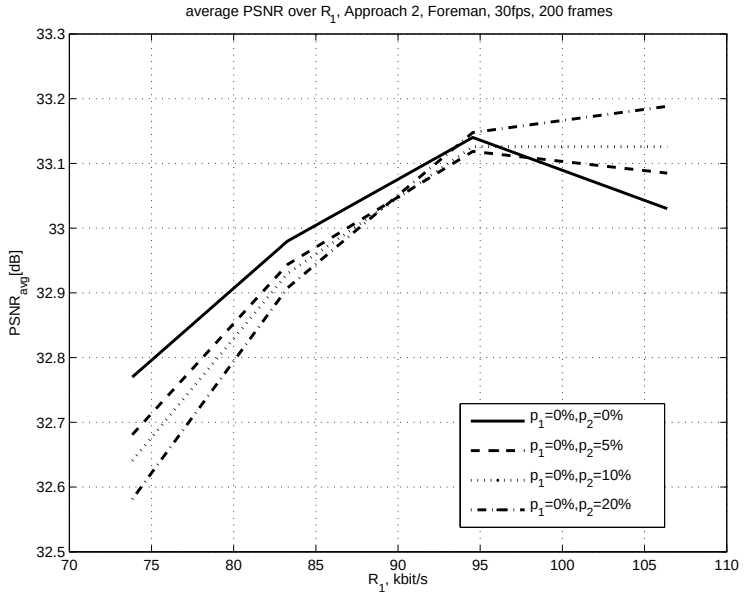


Figure 3.86: $PSNR_{avg}$ over R_1, Approach 2, Foreman with GOB-intra-updates, first stream lossless

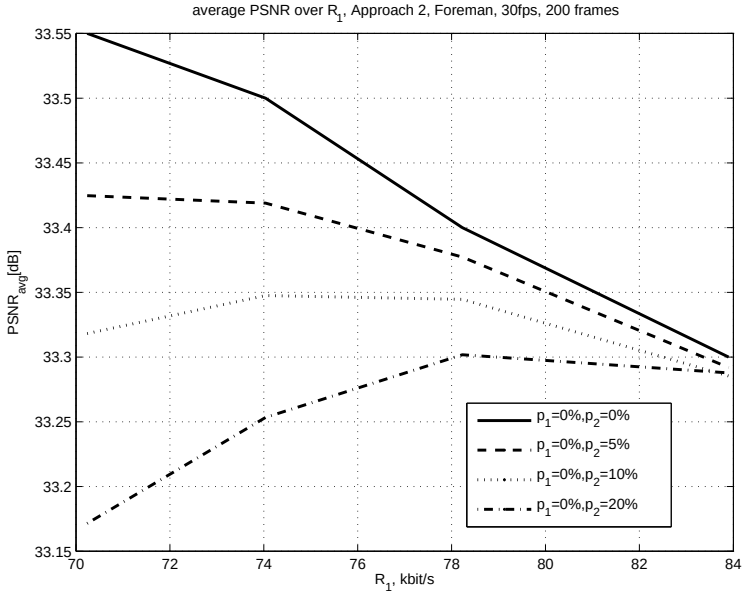


Figure 3.87: $PSNR_{avg}$ over R_1, Approach 2, Foreman with frame-intra-updates, first stream lossless

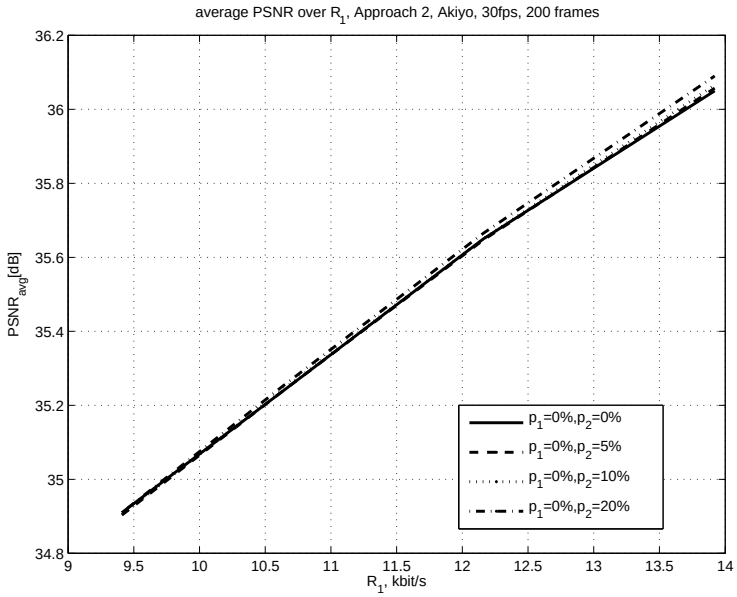


Figure 3.88: $PSNR_{avg}$ over R_1, Approach 2, Akiyo, first stream lossless

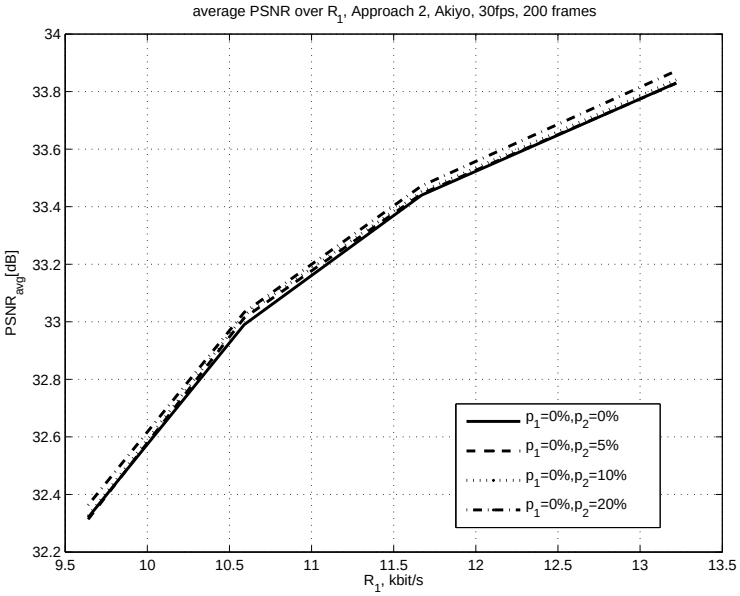


Figure 3.89: $PSNR_{avg}$ over R_1, Approach 2, Akiyo with GOB-intra-updates, first stream lossless

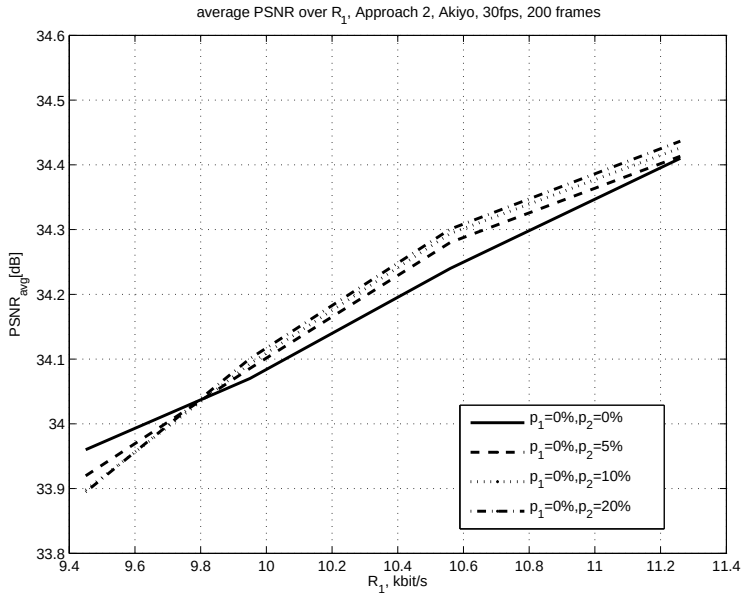


Figure 3.90: $PSNR_{avg}$ over R_1, Approach 2, Akiyo with frame-intra-updates, first stream lossless

case of Foreman, balanced loss probabilities call for balanced rate allocations whether intra-coding is used or not. Only the case with GOB-intra-updates tolerates unbalanced rate allocations. For Akiyo however, unbalance can increase $PSNR_{avg}$ even in the lossless case if Approach 2 is used. The $PSNR_{avg}$ values at different loss probabilities and coding options are summarized in Tables 3.11 and 3.12 for Approach 2 at balanced rate allocations.

The performance of Approach 2, at unbalanced loss probabilities is demonstrated in Figures 3.97, 3.98 and 3.99 for Foreman and 3.100, 3.101 and 3.102 for Akiyo. System performance is enhanced for Foreman when coded with intra updates at unbalanced rate allocation. There is no performance increase at pure inter-coding. Similar to Approach 1, when frame-intra-updates are used, the length of the sequence coded without intras is small so that the high rate stream can help the low rate stream to recover from losses. But as the sequence length increases the interaction between streams increases and we need in that case also the low rate stream to help the high rate stream to recover. However, through the increased effect of the high rate stream on the low rate one, the disadvantage of unbalancing is lessened. This is an extra point, since we sometimes are forced to unbalance the rate allocation, e.g.

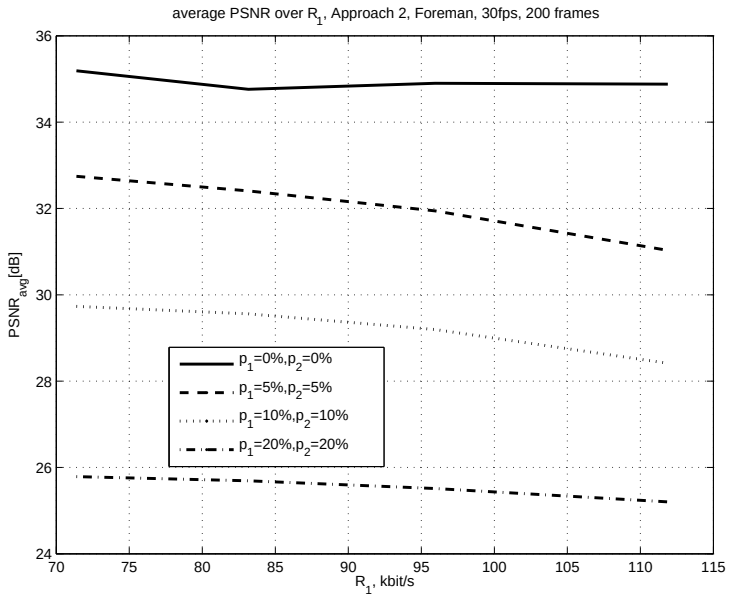


Figure 3.91: $PSNR_{avg}$ over R_1, Approach 2, Foreman, balanced loss probabilities

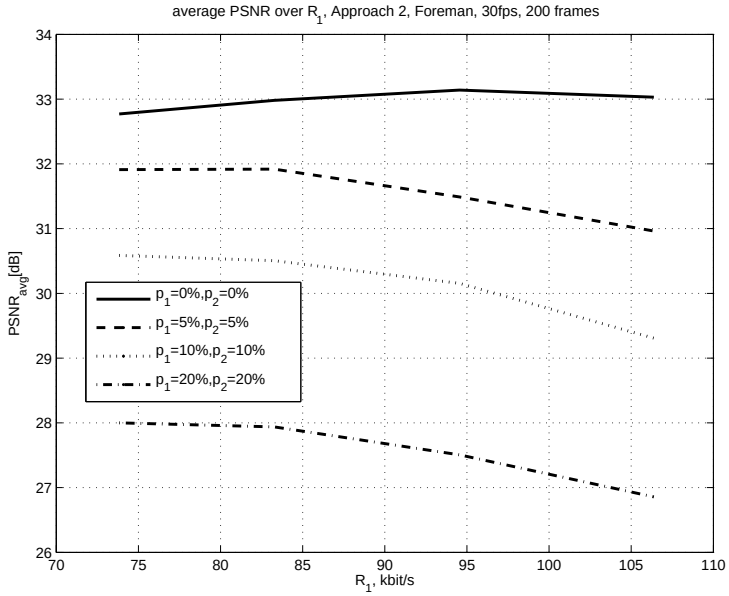


Figure 3.92: $PSNR_{avg}$ over R_1, Approach 2, Foreman with GOB-intra-updates, balanced loss probabilities

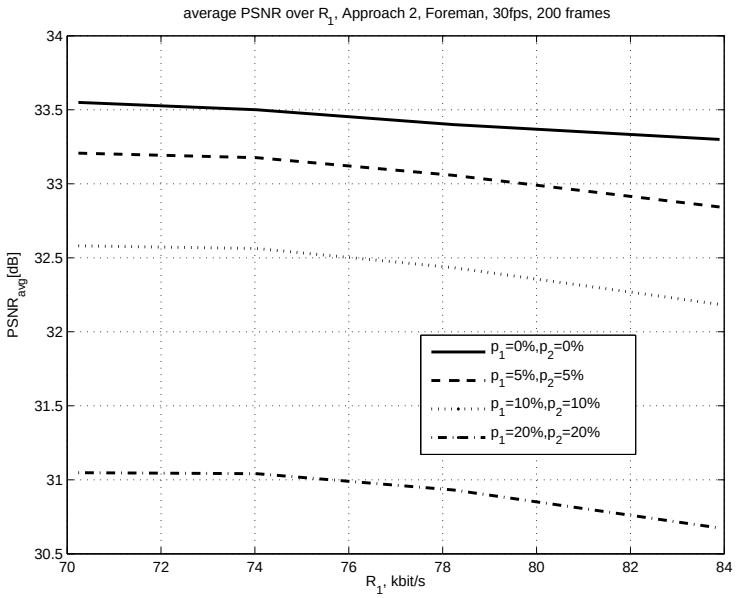


Figure 3.93: $PSNR_{avg}$ over R_1, Approach 2, Foreman with frame-intra-updates, balanced loss probabilities

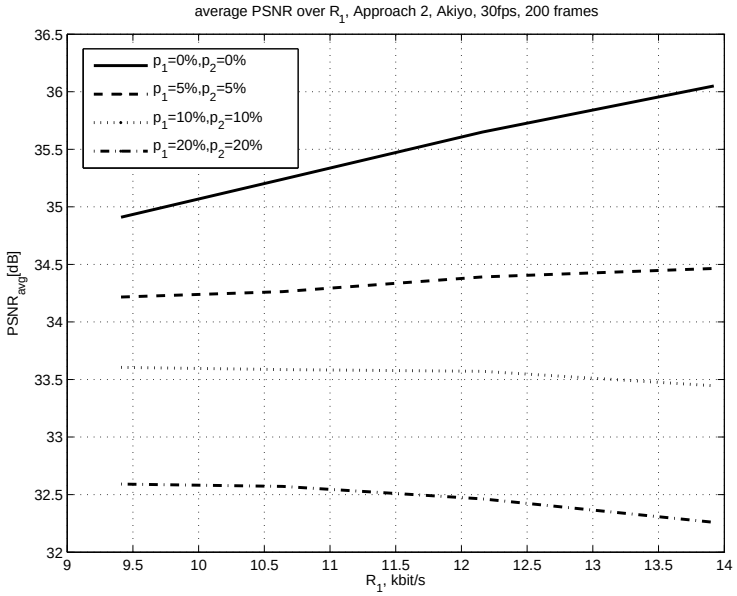


Figure 3.94: $PSNR_{avg}$ over R_1, Approach 2, Akiyo, balanced loss probabilities

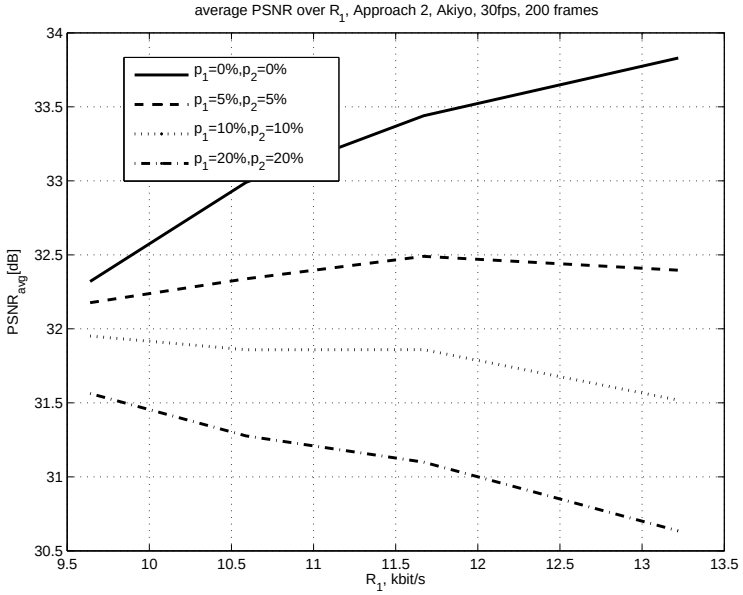


Figure 3.95: $PSNR_{avg}$ over R_1, Approach 2, Akiyo with GOB-intra-updates, balanced loss probabilities

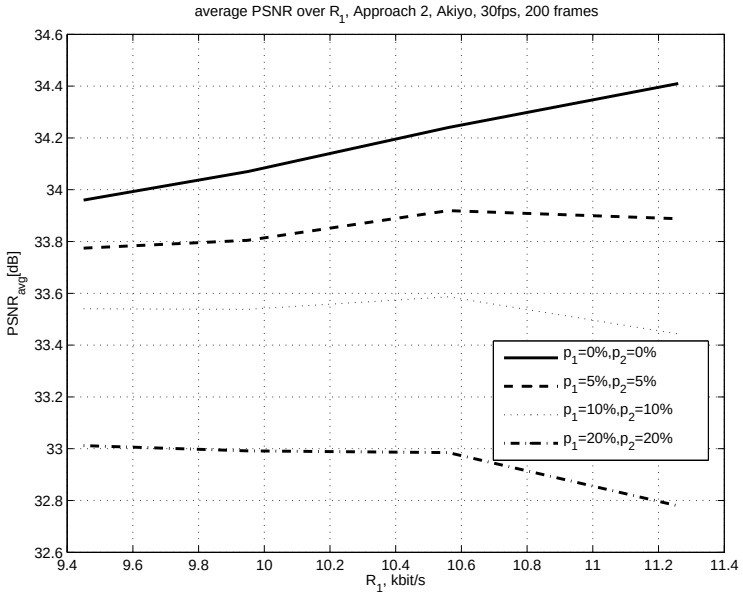


Figure 3.96: $PSNR_{avg}$ over R_1, Approach 2, Akiyo with frame-intra-updates, balanced loss probabilities

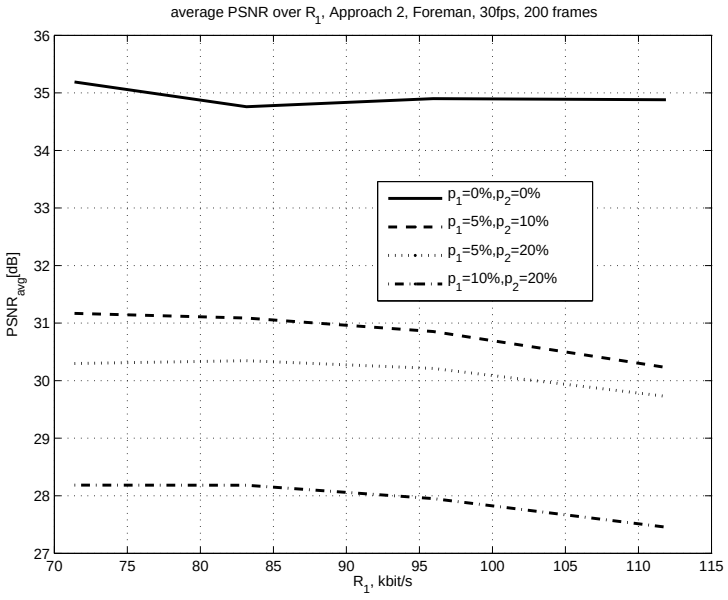


Figure 3.97: $PSNR_{avg}$ over R_1, Approach 2, Foreman, unbalanced loss probabilities

	0%,0%	**0%,5%**	**0%,10%**	**0%,20%**	**5%,5%**
F.	35.19	34.57	34.43	34.30	32.74
F. i.-GOB	32.77	32.68	32.64	32.58	31.91
F. i.-frames	33.55	33.42	33.32	33.17	33.21
A.	34.91	34.90	34.91	34.91	34.22
A. i.-GOB	32.32	32.31	32.33	32.36	32.18
A. i.-frames	33.96	33.92	33.90	33.89	33.77

Table 3.11: MSVC, Approach 2, $PSNR_{avg}$ depending on (p_1, p_2), Part 1

because of bandwidth limitations. For Akiyo however, unbalanced MSVC combined with Approach 2 always pays off, independent of the coding option used.

3.6 Comparison of Approach 1 and Approach 2

Figures 3.103, 3.104 and 3.105 show the comparison of Approach 1 to Approach 2 at different coding options for Foreman and similarly Figures 3.106, 3.107 and 3.108 for Akiyo. For Foreman, Approach 1 is slightly better than Approach 2 at balance condition. This is due to suboptimality of Approach 2, i.e. error propagation after frame interpolation is counted for with an algorithm of depth N. But as the

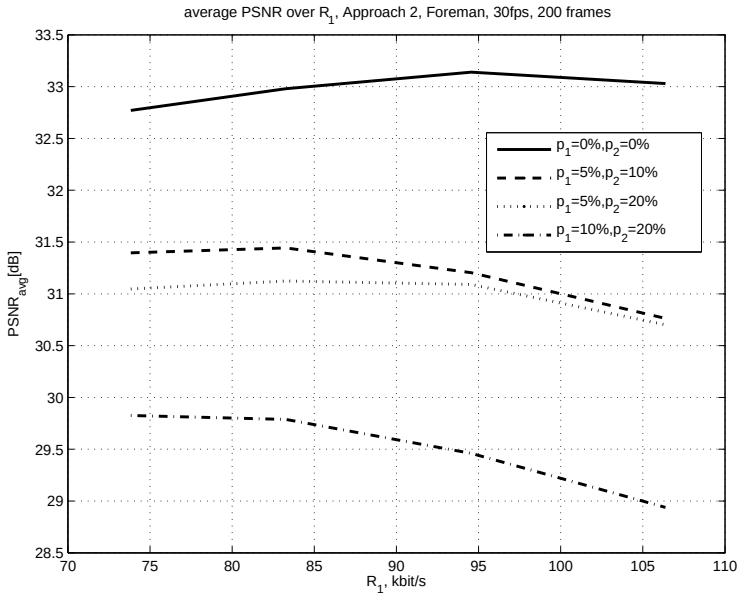


Figure 3.98: $PSNR_{avg}$ over R_1, Approach 2, Foreman with GOB-intra-updates, unbalanced loss probabilities

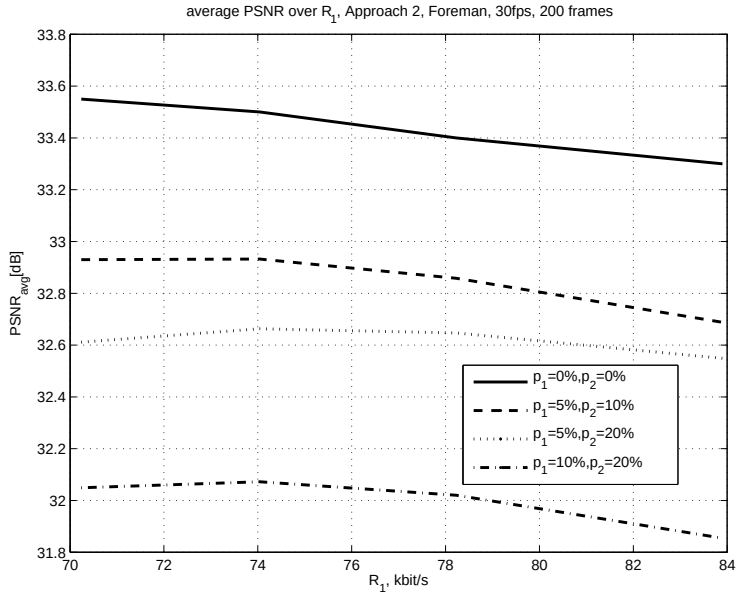


Figure 3.99: $PSNR_{avg}$ over R_1, Approach 2, Foreman with frame-intra-updates, unbalanced loss probabilities

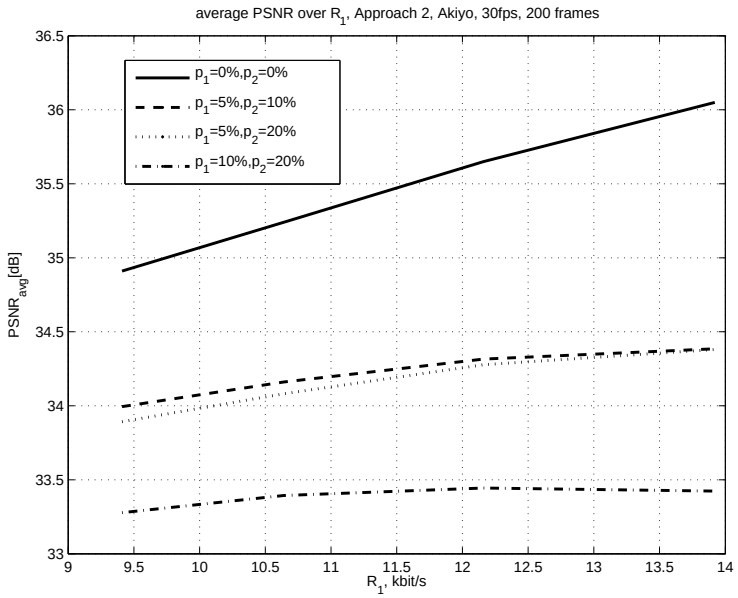


Figure 3.100: $PSNR_{avg}$ over R_1, Approach 2, Akiyo, unbalanced loss probabilities

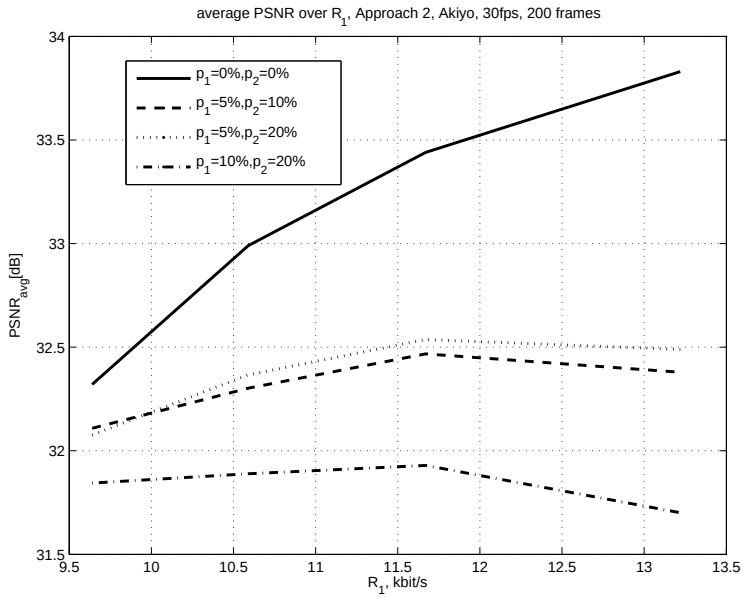
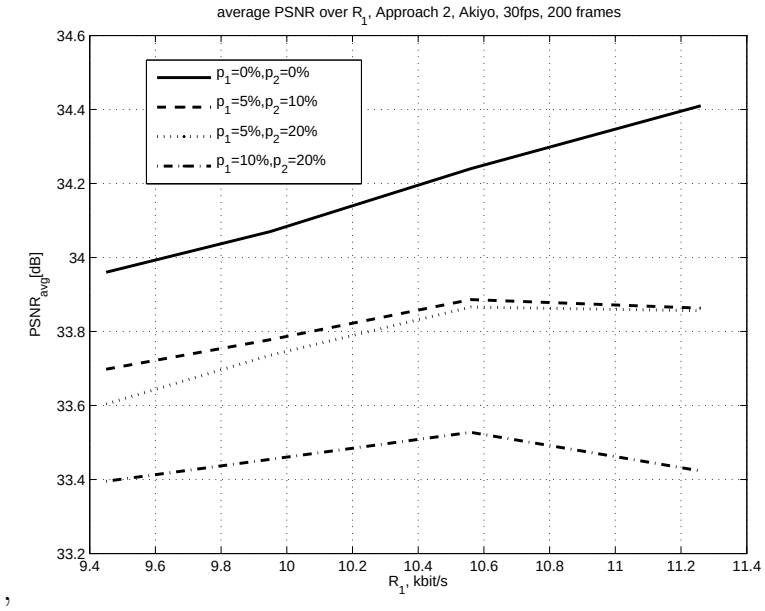


Figure 3.101: $PSNR_{avg}$ over R_1, Approach 2, Akiyo with GOB-intra-updates, unbalanced loss probabilities

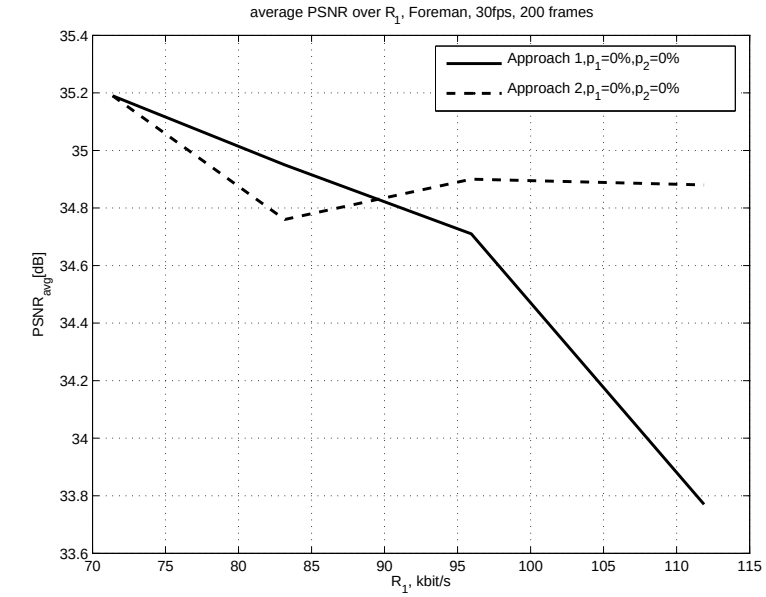


,

Figure 3.102: $PSNR_{avg}$ over R_1, Approach 2, Akiyo with frame-intra-updates, unbalanced loss probabilities

	5%,10%	**5%,20%**	**10%,10%**	**10%,20%**	**20%,20%**
F.	31.17	30.30	29.73	28.18	25.79
F. i.-GOB	31.40	31.05	30.59	29.83	28.00
F. i.-frames	32.93	32.61	32.58	32.05	31.05
A.	33.99	33.89	33.61	33.28	32.59
A. i.-GOB	32.11	32.08	31.95	31.84	31.56
A. i.-frames	33.70	33.60	33.54	33.40	33.01

Table 3.12: MSVC, Approach 2, $PSNR_{avg}$ depending on (p_1, p_2), Part 2



,

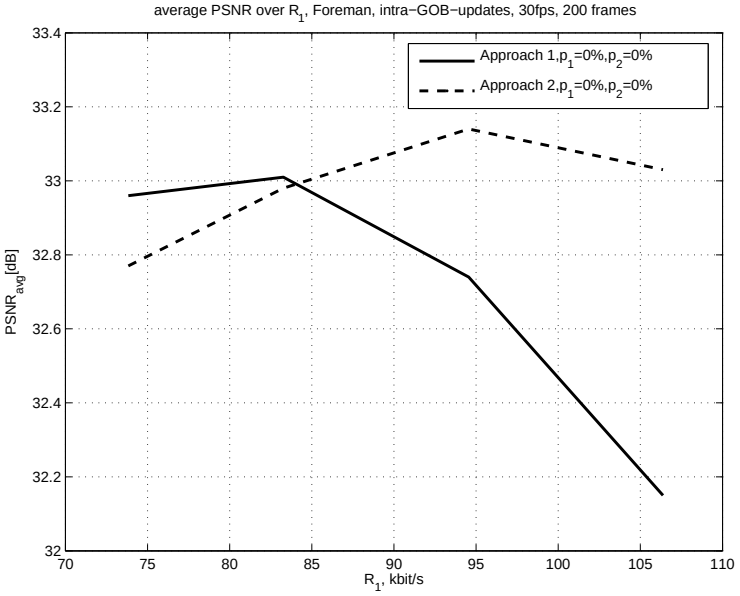
Figure 3.103: $PSNR_{avg}$ over R_1, Comparison of Approach 1 with Approach 2, Foreman

unbalance increases, Approach 2 outperforms Approach 1. The maximum $PSNR_{avg}$ is still obtained at the balanced case when both streams are received lossless. The shift of optimum operating point towards right occurs due to the slight unbalance of R_1 and R_2 even in the most balanced situation (see Table 3.2).

For Akiyo, however, because of the good interpolation ability, combination of unbalance with Approach 2 helps to enhance the performance. As expected from the coding theory Approach 1 obtains the maximum point at balanced condition.

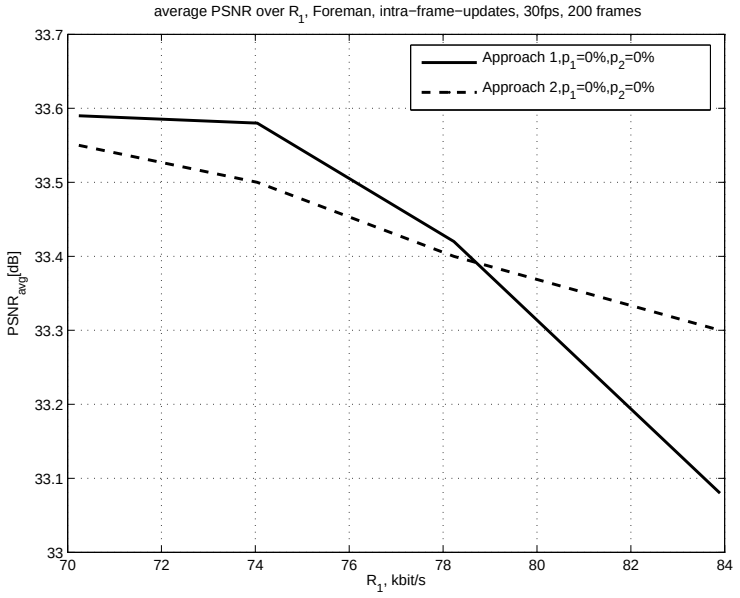
The next question directly arising is how Approach 2 compares to Approach 1 in terms of $PSNR_{avg}$ and in dependence of unbalance. Figures 3.108, 3.109 and 3.110 show the comparison results for Foreman when $p_1 = 0\%$ and p_2 varies between 5% and 20%. Approach 2 supplies higher $PSNR_{avg}$ at every rate allocation. The difference increases with increasing unbalance.

Figures 3.112, 3.113 3.114 show the balanced loss probabilities case with $p_1 = 5\%, p_2 = 5\%$ and $p_1 = 10\%, p_2 = 10\%$ respectively. Approach 2 brings about 0.5 dB advantage at 5% losses, 1 dB at 10% and 20%. The advantage of Approach 2 at unbalanced loss probabilities is even more apparent. As seen from Figures 3.115, 3.116 and 3.117 for $p_1 = 5\%, p_2 = 10\%$, $p_1 = 5\%, p_2 = 20\%$ and $p_1 = 10\%, p_2 = 20\%$



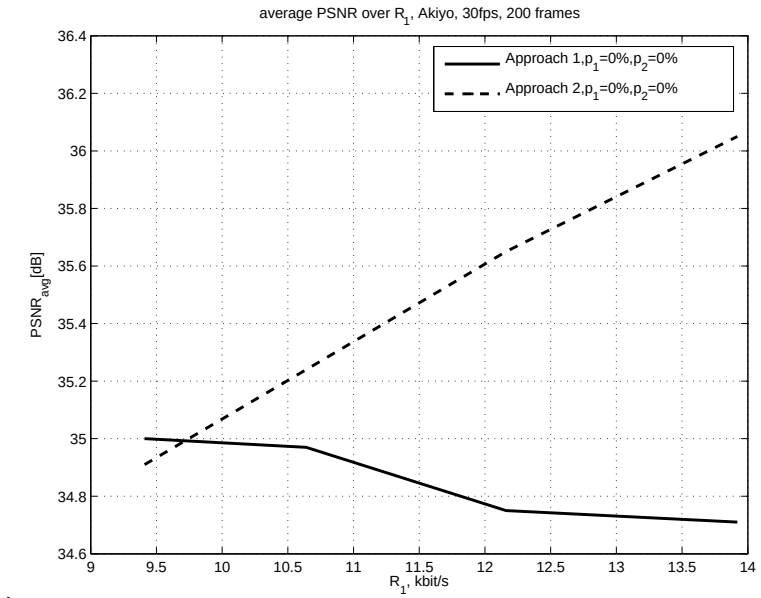
,

Figure 3.104: $PSNR_{avg}$ over R_1, Comparison of Approach 1 with Approach 2, Foreman with GOB-intra-updates



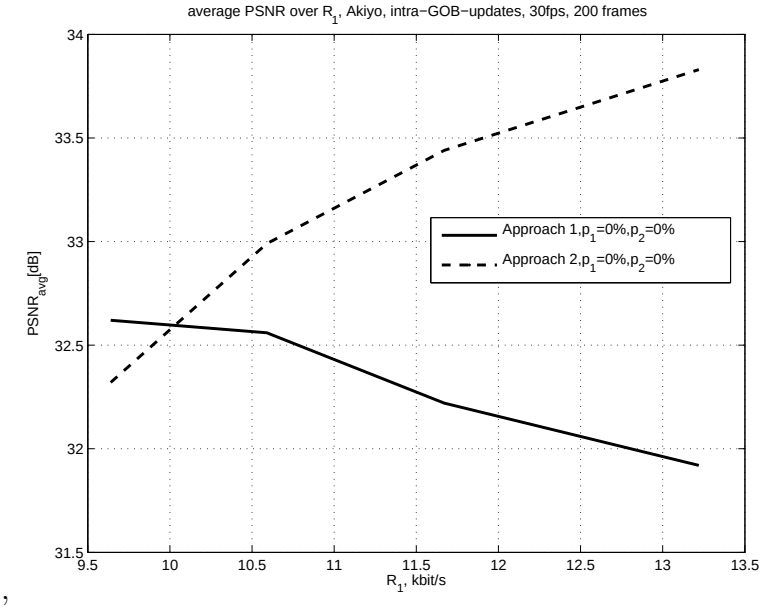
,

Figure 3.105: $PSNR_{avg}$ over R_1, Comparison of Approach 1 with Approach 2, Foreman with frame-intra-updates



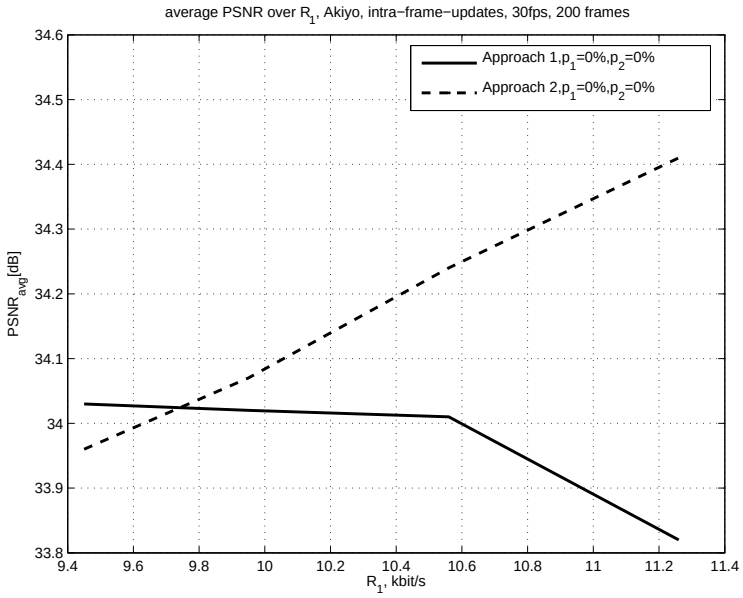
,

Figure 3.106: $PSNR_{avg}$ over R_1, Comparison of Approach 1 with Approach 2, Akiyo



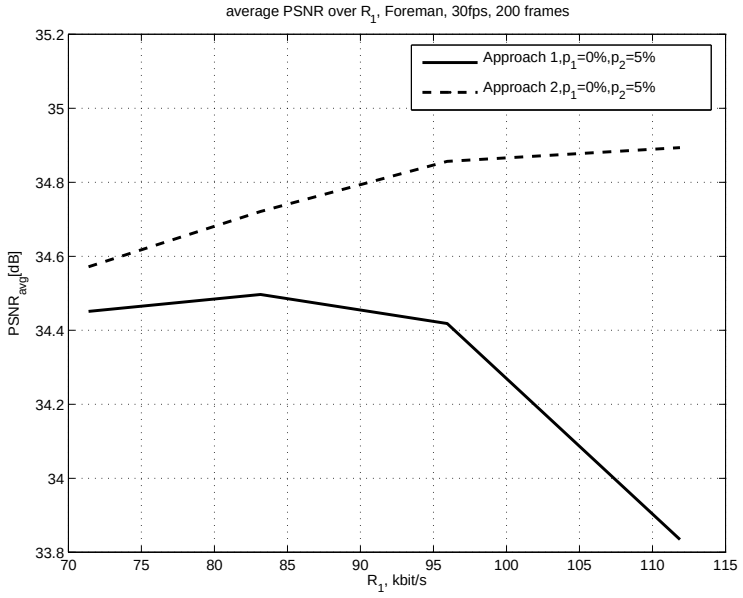
,

Figure 3.107: $PSNR_{avg}$ over R_1, Comparison of Approach 1 with Approach 2, Akiyo with GOB-intra-updates



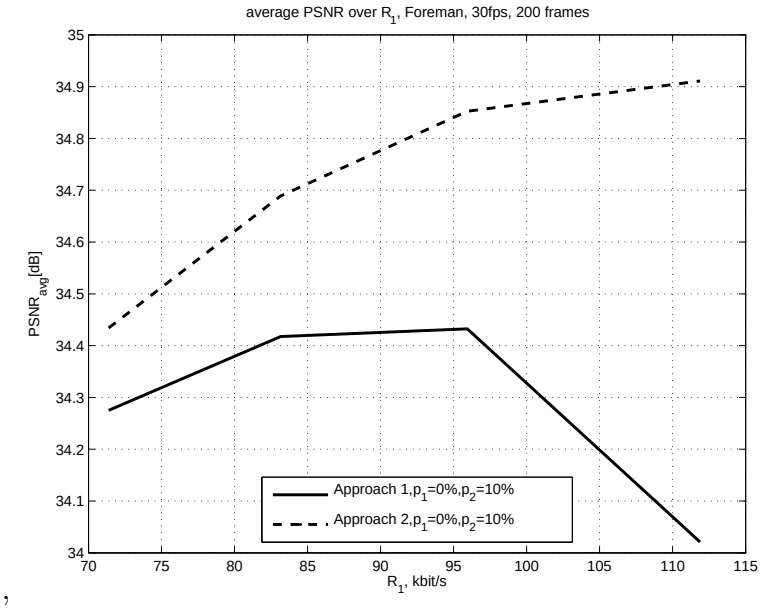
,

Figure 3.108: $PSNR_{avg}$ over R_1, Comparison of Approach 1 with Approach 2, Akiyo with frame-intra-updates



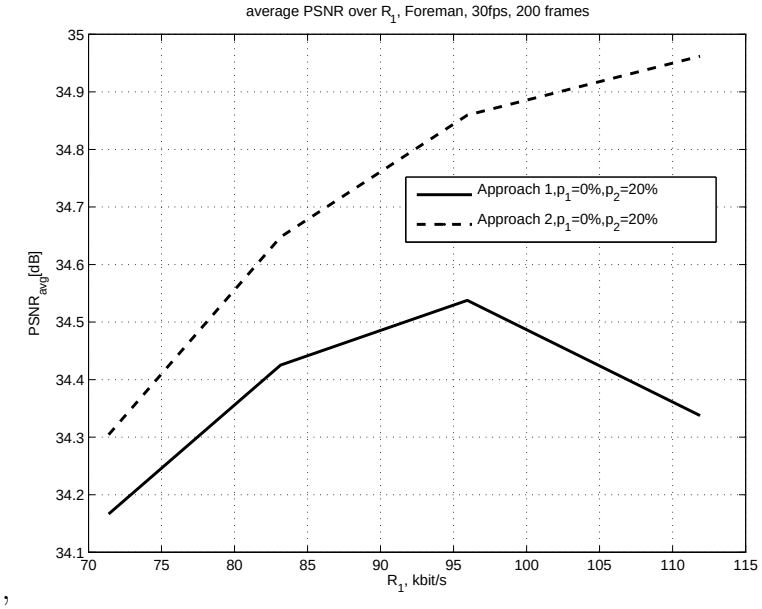
,

Figure 3.109: $PSNR_{avg}$ over R_1, Comparison of Approach 1 with Approach 2, Unbalanced, Foreman, first stream lossless



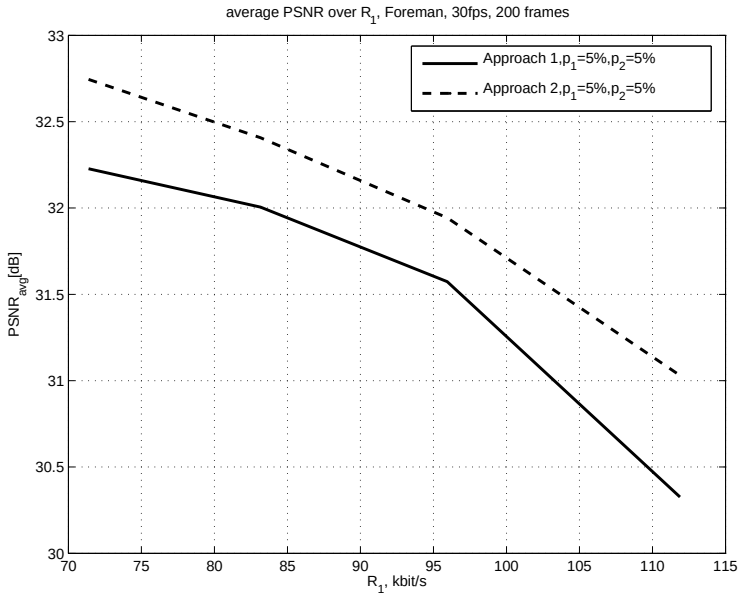
,

Figure 3.110: $PSNR_{avg}$ over R_1, Comparison of Approach 1 with Approach 2, Unbalanced, Foreman, first stream lossless



,

Figure 3.111: $PSNR_{avg}$ over R_1, Comparison of Approach 1 with Approach 2, Unbalanced, Foreman, first stream lossless



,

Figure 3.112: $PSNR_{avg}$ over R_1, Comparison of Approach 1 with Approach 2, Unbalanced, Foreman, balanced loss probabilities

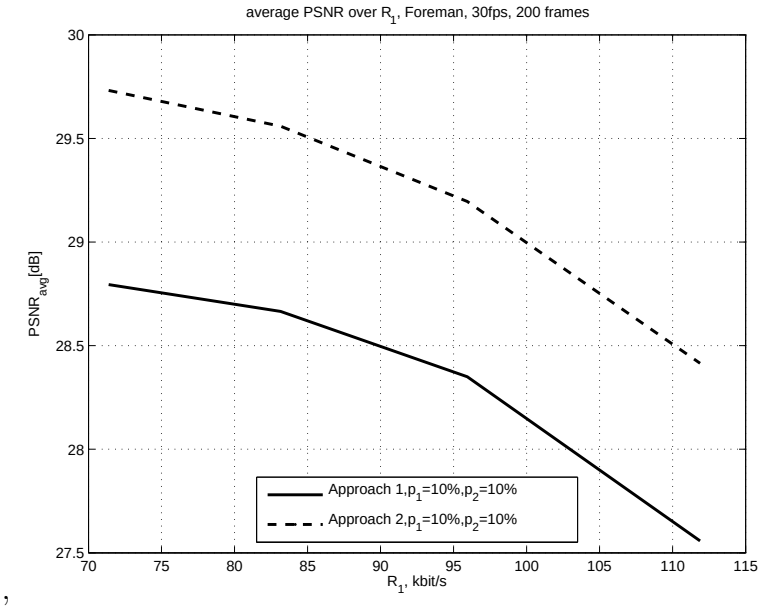
respectively, as the unbalance in loss probabilities and in bitrate allocation increases, the gain of Approach 2 over Approach 1 increases varying between 0.6 dB and 1 dB.

Similar observations are made for the low rate stream Akiyo. Figures 3.118, 3.119and 3.120 show the cases that the first stream is lossless and the second one is lossy with increasing p_2. At balanced bitrate allocation the gain is again about 0.1 dB. The difference is that Approach 1 yields higher $PSNR_{avg}$ with increasing unbalance as observed before.

On the other hand when we have balanced loss probabilities, Approach 1 performs best at equal rate allocation, whereas Approach 2 favors unbalanced operation. The gap between the two increases with increasing unbalance.

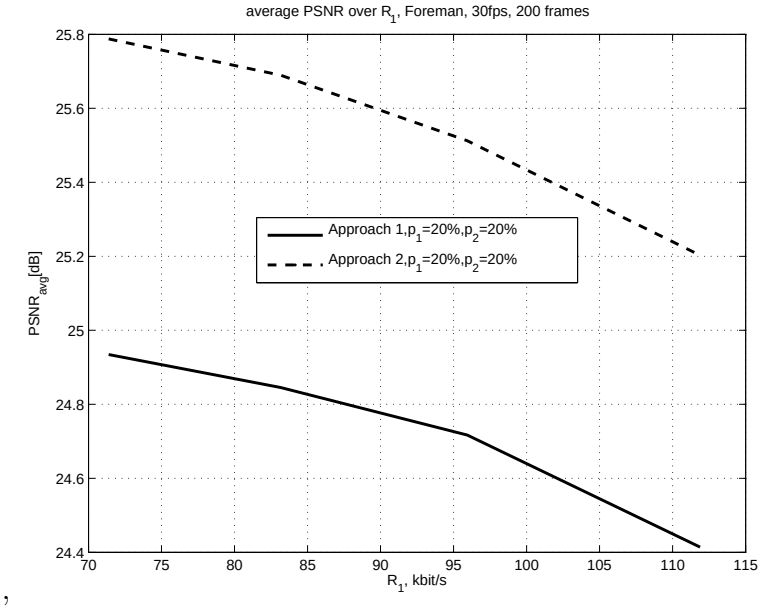
Figures 3.124, 3.125 and 3.126 depict the unbalanced loss probabilities case for Akiyo. The observation is the same as for Foreman.

The frame PSNR over frame number is depicted for a comparison of Approach 1 to Approach 2 at balanced loss probabilities in Figure 3.127 and at unbalanced in 3.128 for Foreman. As the loss rate and the frame number increases the gain of Approach 2 grows. Similar observation but with smaller gain is given for Akiyo in Figures 3.129 and 3.130.



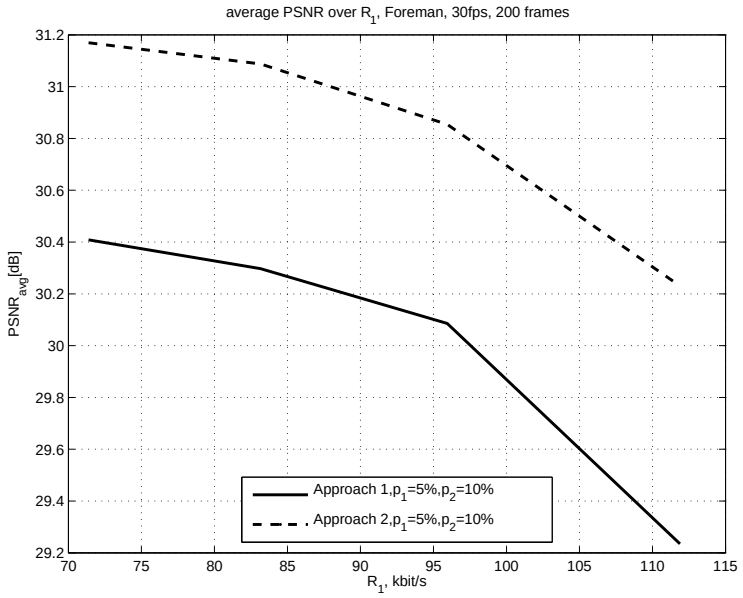
,

Figure 3.113: $PSNR_{avg}$ over R_1, Comparison of Approach 1 with Approach 2, Unbalanced, Foreman, balanced loss probabilities



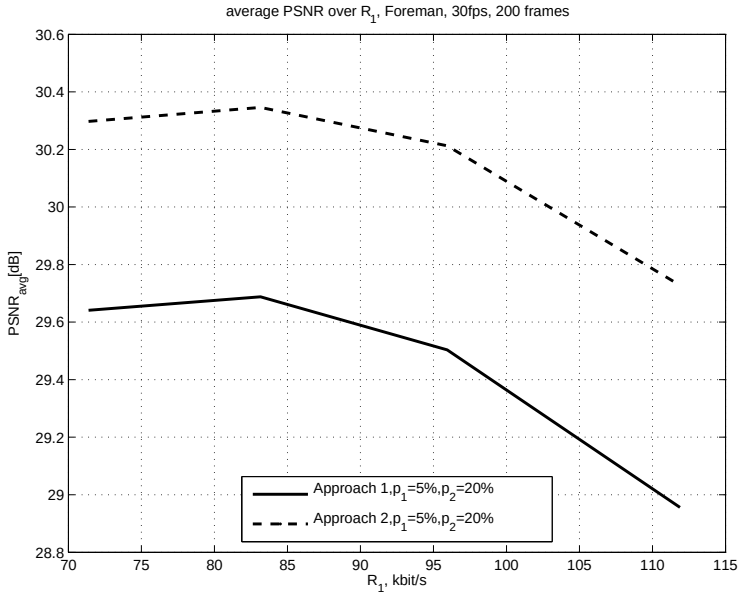
,

Figure 3.114: $PSNR_{avg}$ over R_1, Comparison of Approach 1 with Approach 2, Unbalanced, Foreman, balanced loss probabilities



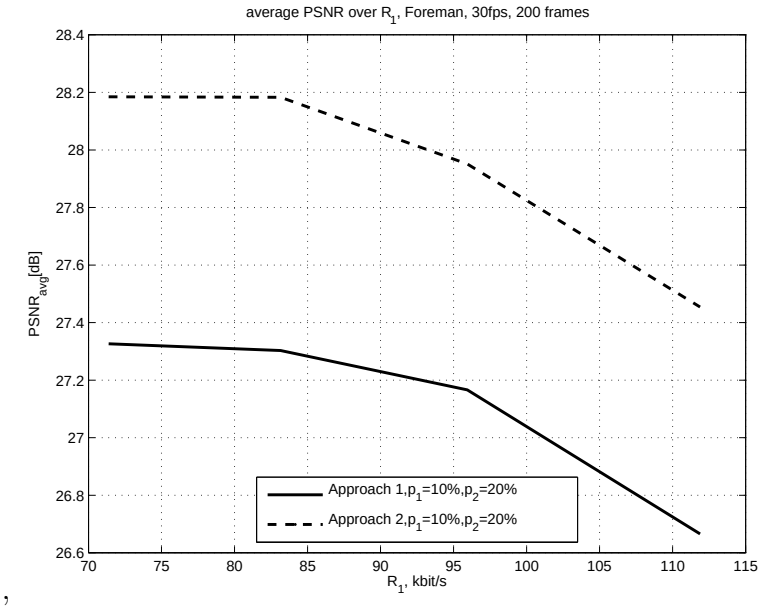
,

Figure 3.115: $PSNR_{avg}$ over R_1, Comparison of Approach 1 with Approach 2, Unbalanced, Foreman, unbalanced loss probabilities



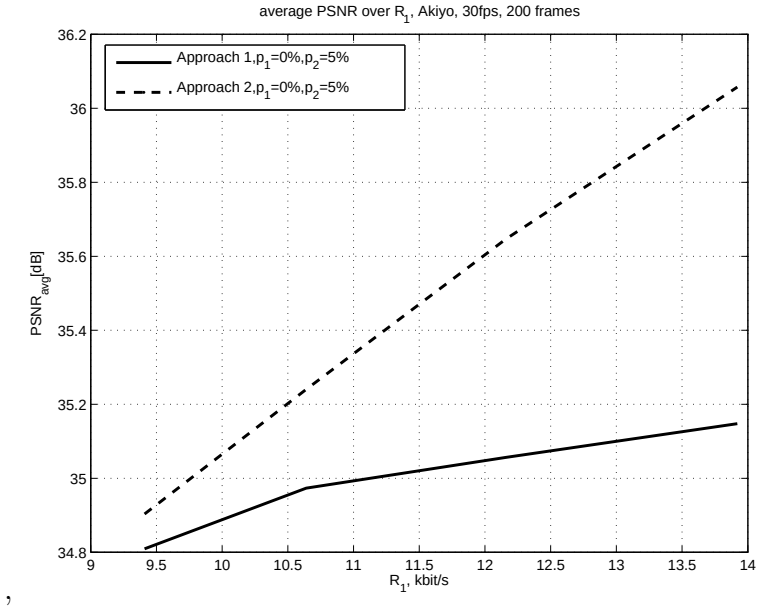
,

Figure 3.116: $PSNR_{avg}$ over R_1, Comparison of Approach 1 with Approach 2, Unbalanced, Foreman, unbalanced loss probabilities



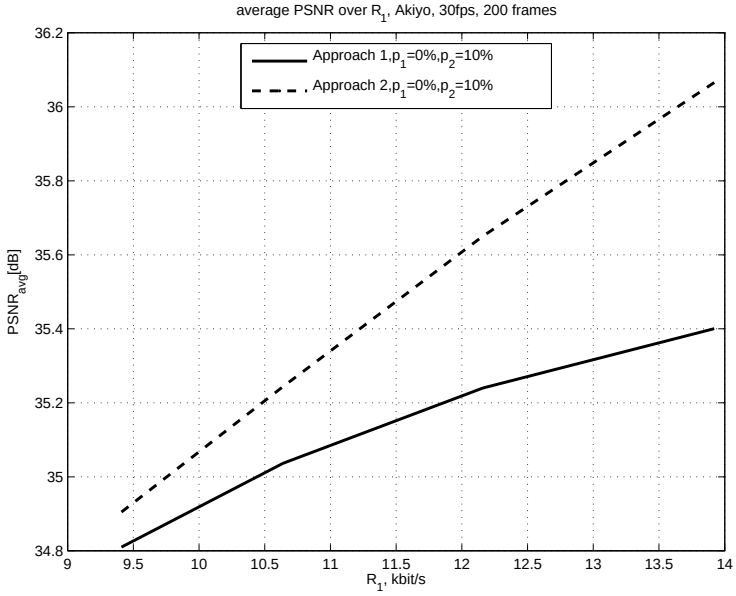
,

Figure 3.117: $PSNR_{avg}$ over R_1, Comparison of Approach 1 with Approach 2, Unbalanced, Foreman, unbalanced loss probabilities



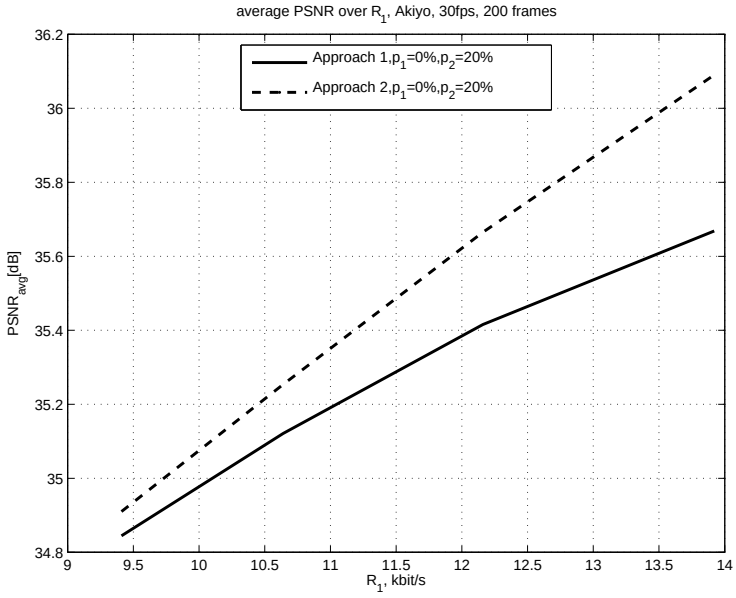
,

Figure 3.118: $PSNR_{avg}$ over R_1, Comparison of Approach 1 with Approach 2, Unbalanced, Akiyo, first stream lossless



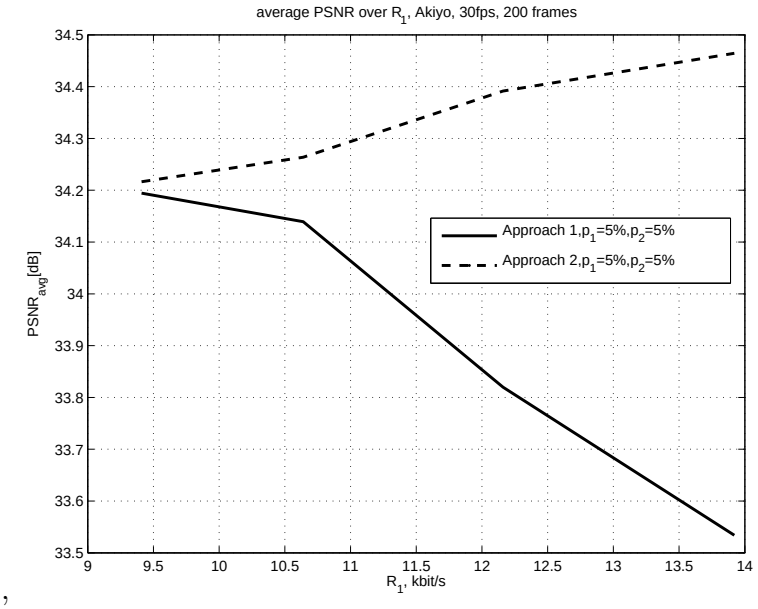
,

Figure 3.119: $PSNR_{avg}$ over R_1, Comparison of Approach 1 with Approach 2, Unbalanced, Akiyo, first stream lossless



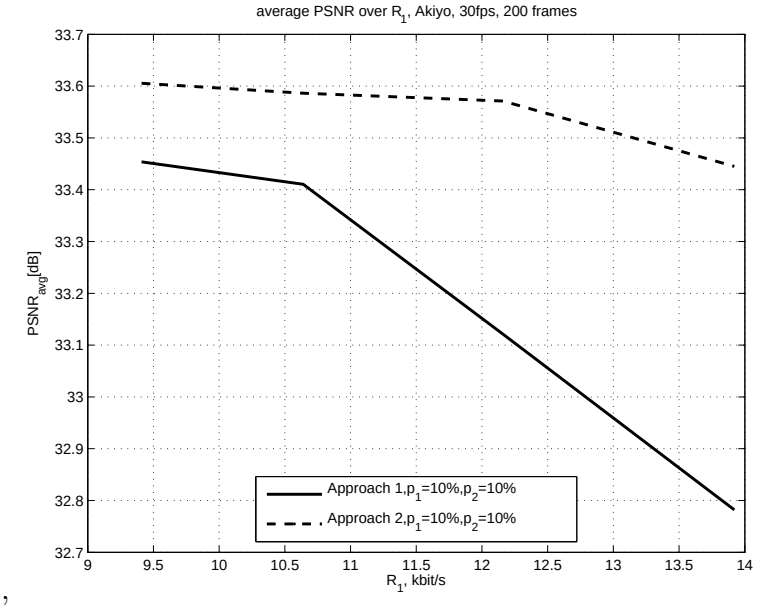
,

Figure 3.120: $PSNR_{avg}$ over R_1, Comparison of Approach 1 with Approach 2, Unbalanced, Akiyo, first stream lossless



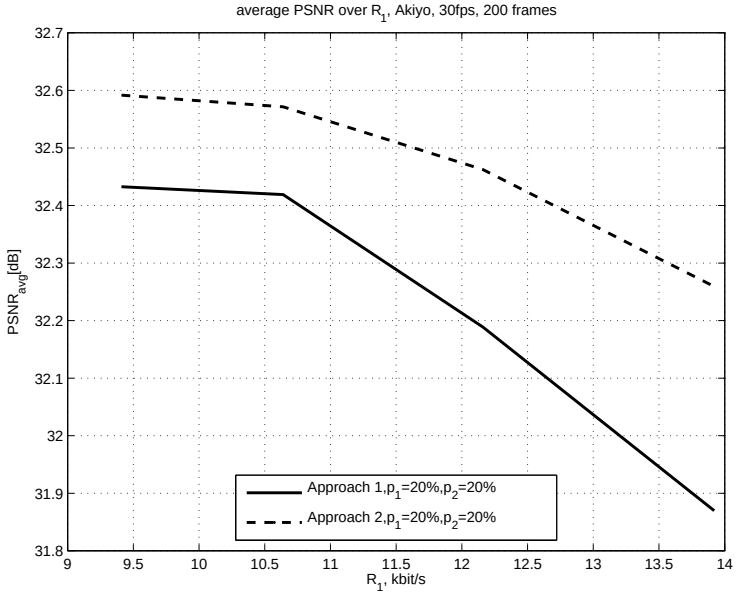
,

Figure 3.121: $PSNR_{avg}$ over R_1, Comparison of Approach 1 with Approach 2, Unbalanced, Akiyo, balanced loss probabilities



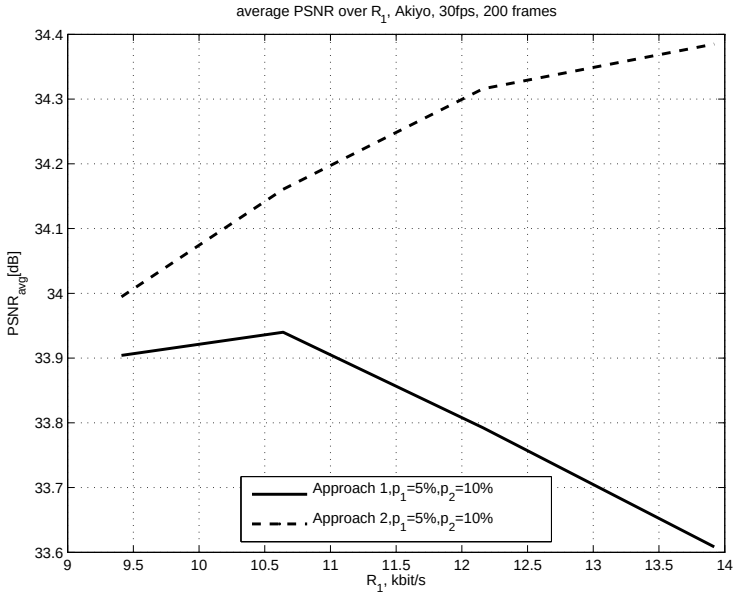
,

Figure 3.122: $PSNR_{avg}$ over R_1, Comparison of Approach 1 with Approach 2, Unbalanced, Akiyo, balanced loss probabilities



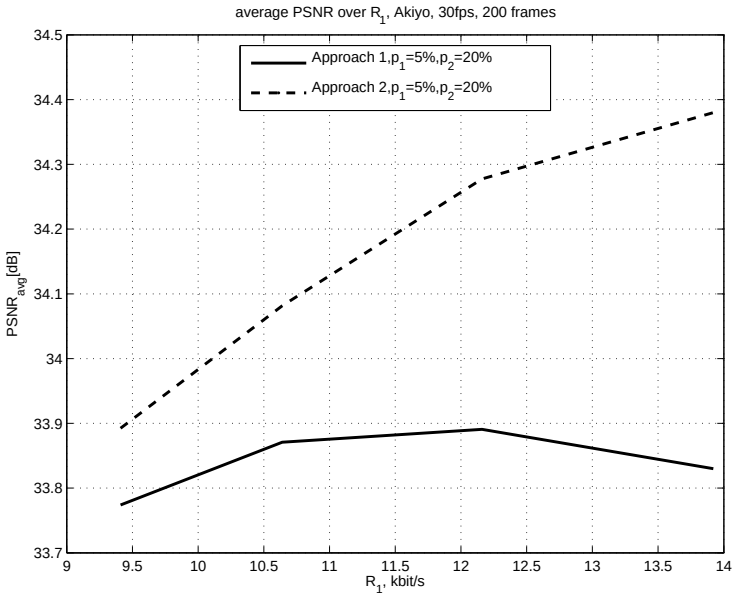
,

Figure 3.123: $PSNR_{avg}$ over R_1, Comparison of Approach 1 with Approach 2, Unbalanced, Akiyo, balanced loss probabilities



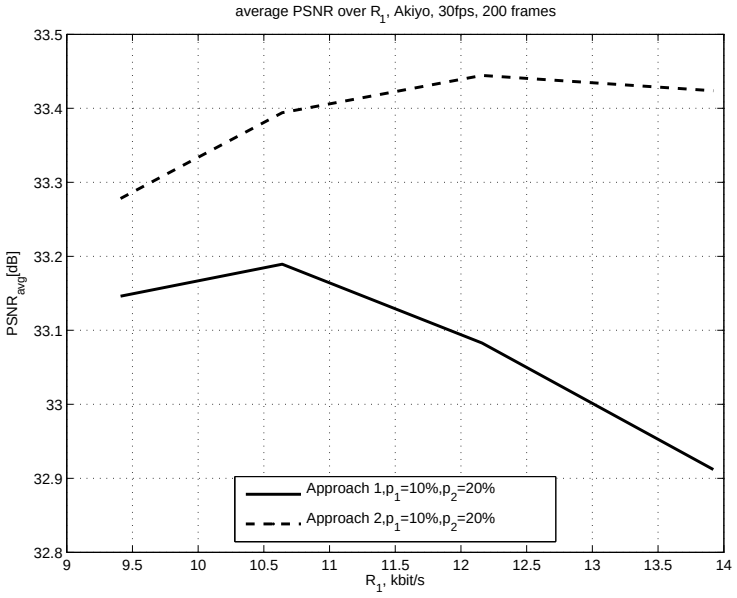
,

Figure 3.124: $PSNR_{avg}$ over R_1, Comparison of Approach 1 with Approach 2, Unbalanced, Akiyo, unbalanced loss probabilities



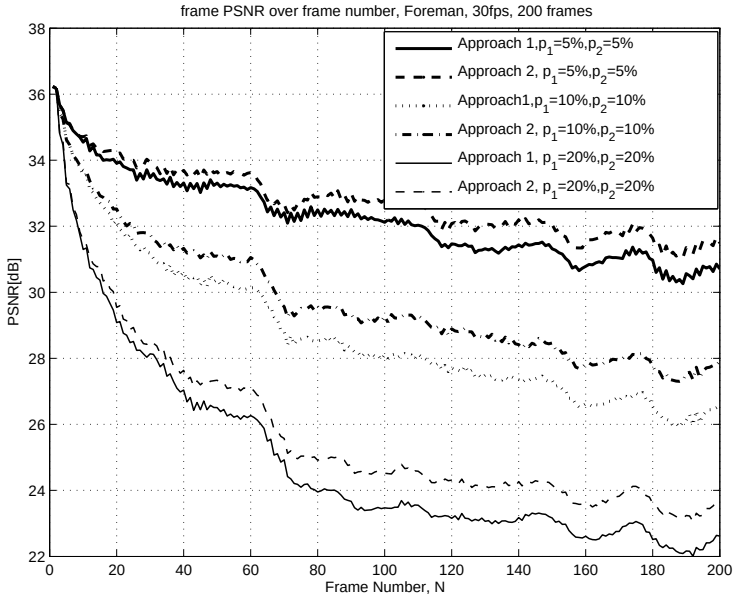
,

Figure 3.125: $PSNR_{avg}$ over R_1, Comparison of Approach 1 with Approach 2, Unbalanced, Akiyo, unbalanced loss probabilities



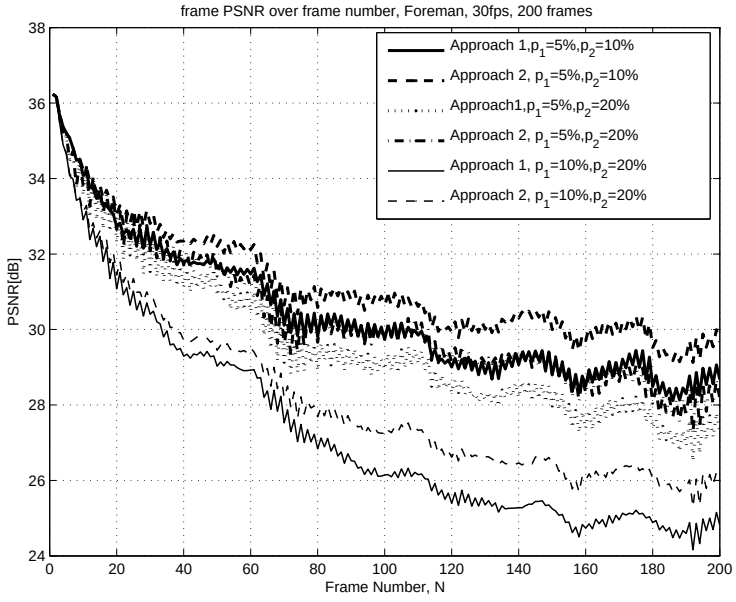
,

Figure 3.126: $PSNR_{avg}$ over R_1, Comparison of Approach 1 with Approach 2, Unbalanced, Akiyo, unbalanced loss probabilities



,

Figure 3.127: PSNR over Frame Number, Comparison of Approach 1 with Approach 2, Unbalanced, Foreman, balanced loss probabilities



,

Figure 3.128: PSNR over Frame Number, Comparison of Approach 1 with Approach 2, Unbalanced, Foreman, unbalanced loss probabilities

To sum up, Approach 2 outperforms Approach 1 almost always. Only in lossless case, it is suboptimal to use Approach 2. Unbalanced loss probabilities call for unbalanced bitrate allocations, especially if combined with Approach 2. If the sequence length coded without intras is shorter, the gain at unbalanced operation is larger. At balanced loss probabilities, balanced rate allocation is to be preferred. Unbalanced operation combined with Approach 2 can only make sense if the motion is low, i.e. interpolation error is small.

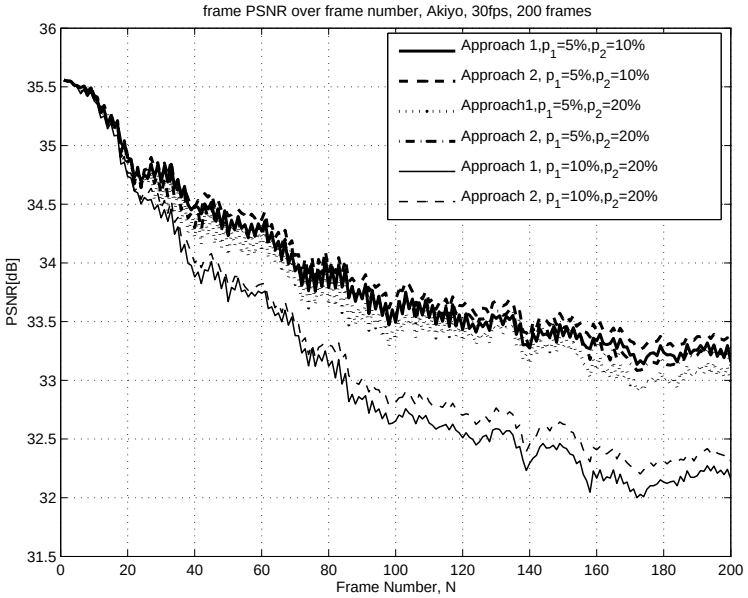


Figure 3.129: PSNR over Frame Number, Comparison of Approach 1 with Approach 2, Unbalanced, Akiyo, balanced loss probabilities

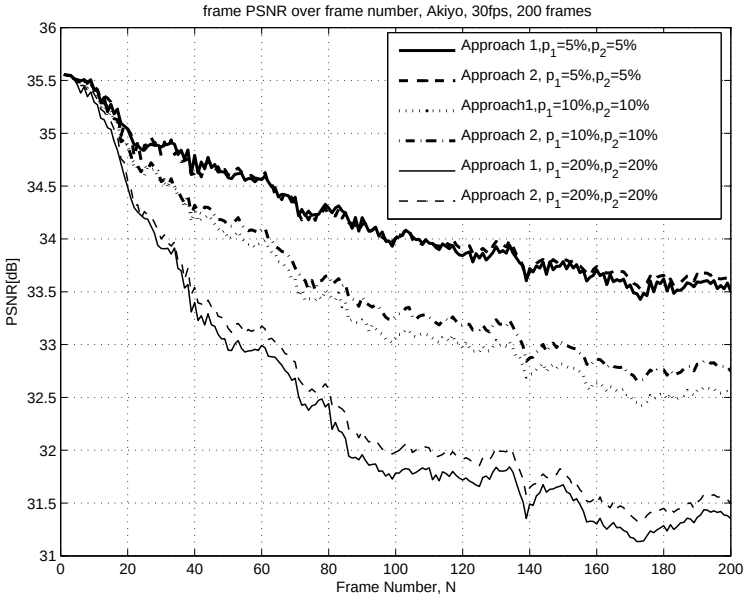


Figure 3.130: PSNR over Frame Number, Comparison of Approach 1 with Approach 2, Unbalanced, Akiyo, unbalanced loss probabilities

Chapter 4

Multi-State vs. Layered and Single-State Coding

In this chapter we will compare MSVC to Single-State (SSC) and Layered Coding (LC). In LC we will assume that the enhancement layer is sent over another path as the base layer. We will concentrate on the Temporal Layered Coding (TLC) where each second frame is coded in the enhancement layer and the rest on the base layer. In SSC, the produced stream is sent over one path. First we will assume that the loss probabilities of the two chosen paths are the same, i.e.: $p_1 = p_2$, but the losses are independent from each other. In the second part, we will compare SSC to MSVC where one of the paths used for MSVC is lossless. The work is partially given in [37] and [35].

Moreover we will investigate two cases: Since each packet contains a frame, when a packet is lost, the associated motion information is also lost. In such a case SSC will use frame repetition to replace the lost frames and MSVC state recovery based on interpolation. In the second case we will assume that the motion vectors for SSC are always received although the associated packets are lost. Actually to ensure that the motion vectors are received, additional bitrate for extra protection such as FEC/ARQ is to be employed, but this aspect will not be considered here. It will be assumed that all the bitrate available is used for coding the sequence.

We use the same total bitrate R_T for all three coding methods. The coding parameters are given in tables 3.2, 4.1 and 4.2 for MSVC, SSVC and TLC respectively. "A." denotes the sequence Akiyo whereas "F." stands for Foreman.

	QP first	QP remain.	i.-GOB period	i.-frame period	Bitrate [kbit/s]
F.	16	16			137.28
F. i.-GOB	16	17	3		136.51
F. i.-frames	16	17		30	133.88
F. + ARQ(3),5%	19	16			137.00
F. + ARQ(3),10%	12	17			118.90
F. + ARQ(3),20%	12	17			116.72
F. i.-frames + ARQ(3),5%	19	16		30	137.00
F. i.-frames + ARQ(3),10%	12	17		30	118.90
F. i.-frames + ARQ(3),20%	20	17		30	116.72
A.	18	17			20.80
A. i.-GOB	23	23	3		20.80
A. i.-frames	23	23		30	18.86
A. + ARQ(3),5%	18	18			18.17
A. + ARQ(3),10%	20	18			18.06
A. + ARQ(3),20%	19	19			15.77
A. i.-frames + ARQ(3),5%	24	23		30	17.79
A. i.-frames + ARQ(3),10%	24	24		30	16.66
A. i.-frames + ARQ(3),20%	25	24		30	16.00

Table 4.1: SSVC Coding Parameters

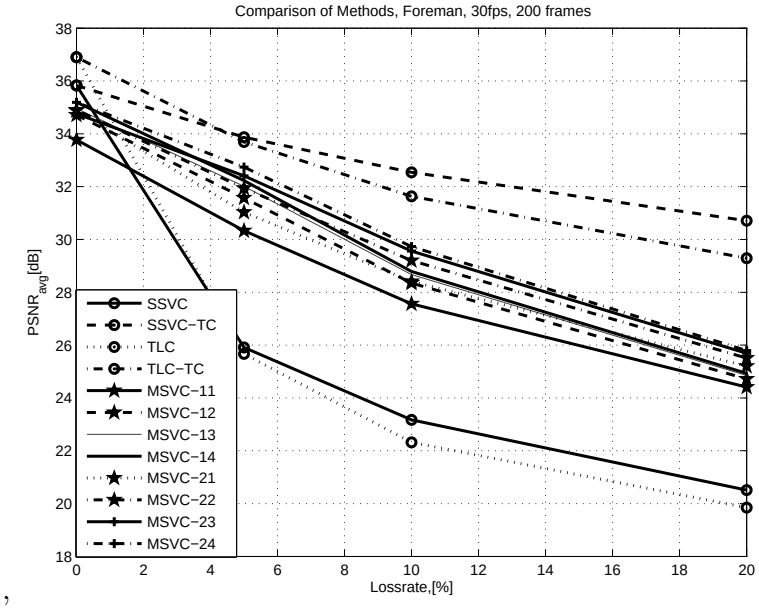
	QP first	QP remain.	i.-GOB period	i.-frame period	Bitrate [kbit/s]
F.	14	14			147.95
F. i.-GOB	15	15	2		142.86
F. i.-frames	15	15		15	143.60
F. + ARQ(3),5%	14	15			135.78
F. + ARQ(3),10%	15	15			126.02
F. + ARQ(3),20%	15	16			116.11
F. i.-frames + ARQ(3),5%	15	16		15	133.59
F. i.-frames + ARQ(3),10%	16	16		15	132.24
F. i.-frames + ARQ(3),20%	16	17		15	114.67
A.	17	17			18.62
A. i.-GOB	22	22	2		19.65
A. i.-frames	22	22		15	20.01
A. + ARQ(3),5%	16	17			18.28
A. + ARQ(3),10%	17	17			18.05
A. + ARQ(3),20%	17	18			16.82
A. i.-frames + ARQ(3),5%	23	23		15	17.79
A. i.-frames + ARQ(3),10%	23	24		15	16.66
A. i.-frames + ARQ(3),20%	24	25		15	16.00

Table 4.2: TLC Coding Parameters

Figure 4.1 compares the coding methods SSVC, SSVC-TC, TLC, TLC-TC and MSVC. SSVC and TLC use the last frame received if the current one is lost. SSVC-TC and TLC-TC, on the other hand, use the motion vectors of the current frame to perform motion compensated error concealment, i.e. the blocks of the reference frame are moved to the positions pointed to by the motion vectors. The block errors constitute the motion compensated difference images (packet contents). As seen from the figure, receiving the motion vectors to be used for error concealment makes a big difference: e.g. 10 dB averaged over 200 coded frames at 20% loss rate. As the loss rate increases this difference increases. Moreover, SSVC performs better than TLC. The difference increases with increasing loss rate although $PSNR_{avg}$ is a bit lower at the lossless case. For MSVC, we test unbalanced rate allocations as well as the two approaches Approach 1 and Approach 2 at symmetric loss rates, i.e. $p_1 = p_2$. Here $MSVC_{xy}$ denotes the x^{th} Approach and the y^{th} rate allocation. 4^{th} rate allocation is the balanced one and the 1^{st} one is the most unbalanced one. As depicted clearly in the figure, $MSVC_{24}$ gives the best performance as expected: Approach 2 yields always better than Approach 1, moreover balanced loss probabilities call for balanced rate allocations. But between MSVC and SSVC-TC there is still a PSNR gap of 1, 3 and 5 dB at 5%, 10% and 20% loss rates respectively. However since we did not count for the additional error protection to guarantee the reception of the motion vectors, this is not a fair comparison. The curve we obtain serves as a kind of target. If we loose the motion information when the corresponding frame is lost, MSVC outperforms SSVC by 5 to 7 dB over the loss rate range. This is a huge gain although we assumed that both channels used for MSVC are error prone.

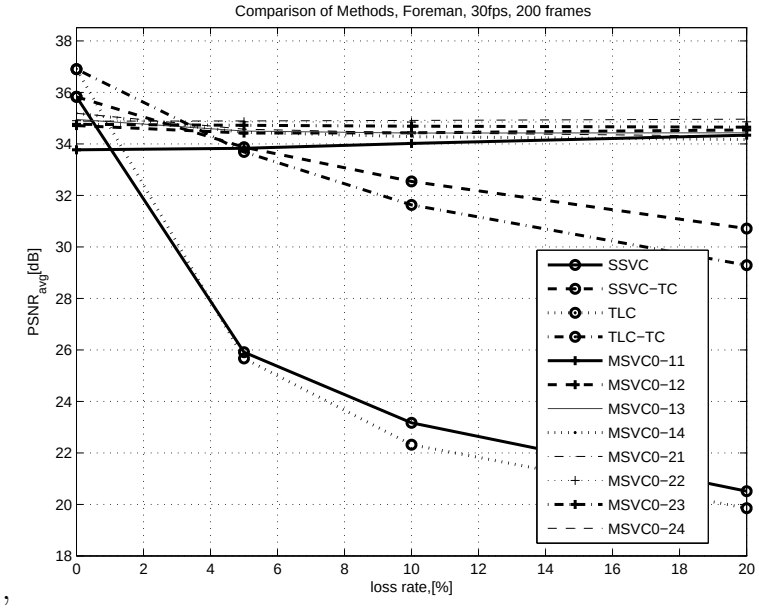
Figure 4.2 performs the same comparison for the case that the first channel used for MSVC is lossless whereas the second one has the same loss rate as the channel used by SSVC and TLC. For TLC we assumed that both base and enhancement layers are sent over lossy channels with the same loss rate. The probability that we catch a second channel with a probable better transmission condition is the main idea behind path diversity. This fact is also confirmed by Figure 4.2. At 20% loss rate, $MSVC_{21}$ outperforms SSVC by 14 dB and even SSVC-TC by 4 dB when $p_1 = 0\%$. $MSVC_{21}$ is the best option in this case: When the first channel is lossless it is the best to allocate more bitrate to the first stream.

Figure 4.3 shows for Akiyo the comparison of methods when both channels are



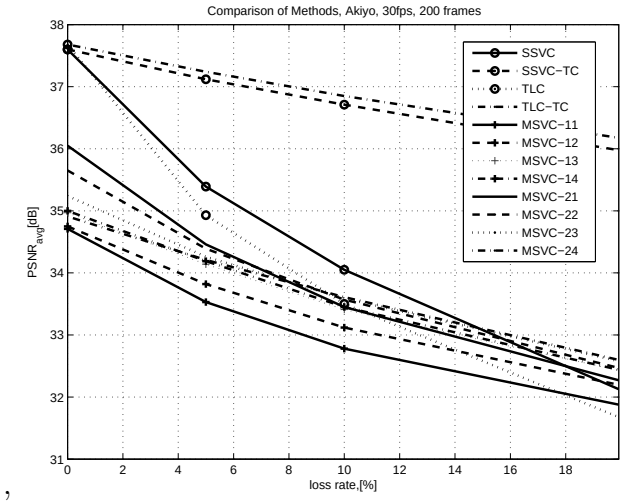
,

Figure 4.1: Comparison of SSVC, SSVC-TC, TLC, TLC-TC and MSVC, all channels have the same loss rate, Foreman.



,

Figure 4.2: Comparison of SSVC, SSVC-TC, TLC, TLC-TC and MSVC, one of the MSVC channels is lossless, Foreman.



,

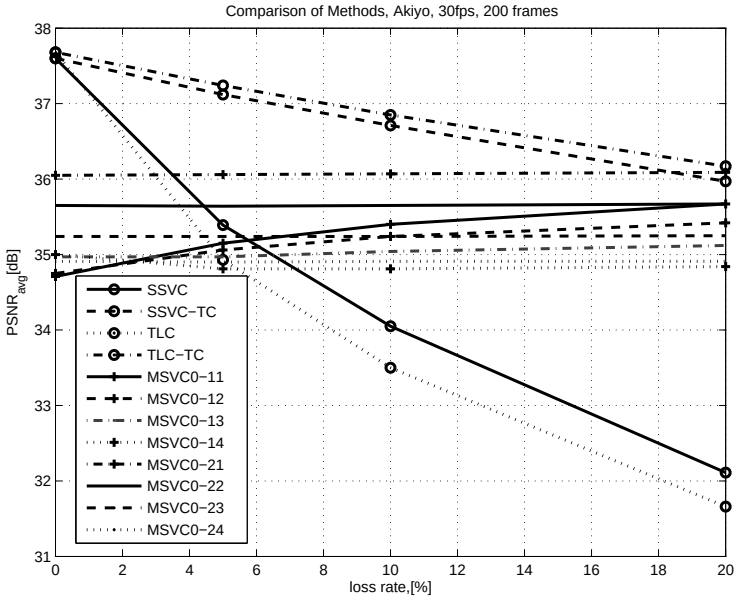
Figure 4.3: Comparison of SSVC, SSVC-TC, TLC, TLC-TC and MSVC, all channels have the same loss rate, Akiyo

lossy with the same loss probability. Error concealment is easier due to low motion. Therefore SSVC with frame repetition as error concealment, gives good performance in lossy environment and outperforms MSVC when loss rate is smaller than about 15%. But when loss rate increases beyond this limit, it is still better to employ MSVC. Although unbalanced rate allocations are better at smaller loss rates ($MSVC_{21}$), larger loss rates require balanced rate distributions ($MSVC_{24}$).

Next we see in Figure 4.4 what happens for Akiyo when the first channel is lossless. The unbalanced $MSVC_{21}$ performs the best among all MSVC options. The '0' in MSVC0 means that the transmission of the first stream was lossless. $MSVC_{21}$ performs 4dB better than SSVC at 20% losses. The performance of SSVC-TC is reached only at larger loss rates.

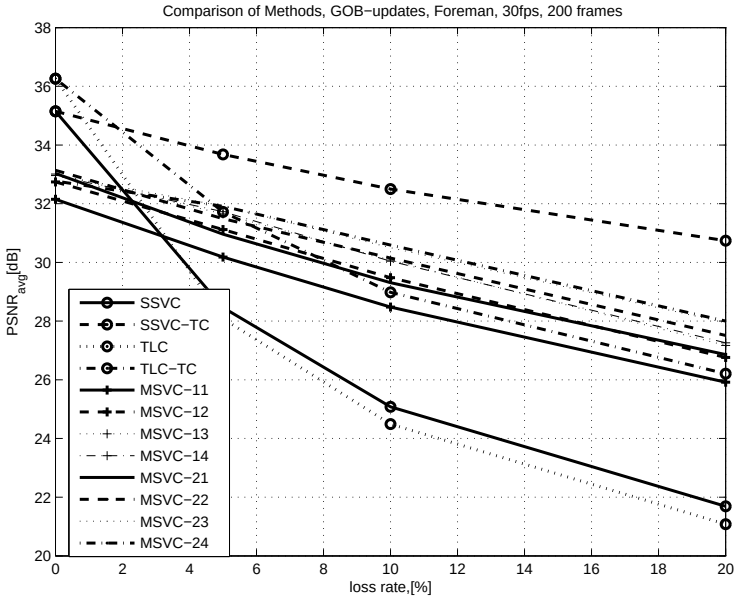
Further, we compare the methods when GOB-intra-updates are used. The coding parameters are listed in tables 4.1 and 4.2. The gap between MSVC and SSVC decreases while between MSVC and SSVC-TC increases (see figure 4.5). Unbalance in bitrate is preferred when the first channel is lossless (see figure 4.6). Again the gaps between coding methods are smaller with GOB-intra-updates.

The effect of decreasing gaps is best presented in 4.7 and 4.8 one for balanced loss probabilities, the other for lossless transmission of the first stream. Unbalanced MSVC performs better than SSVC-TC at lossy environment.



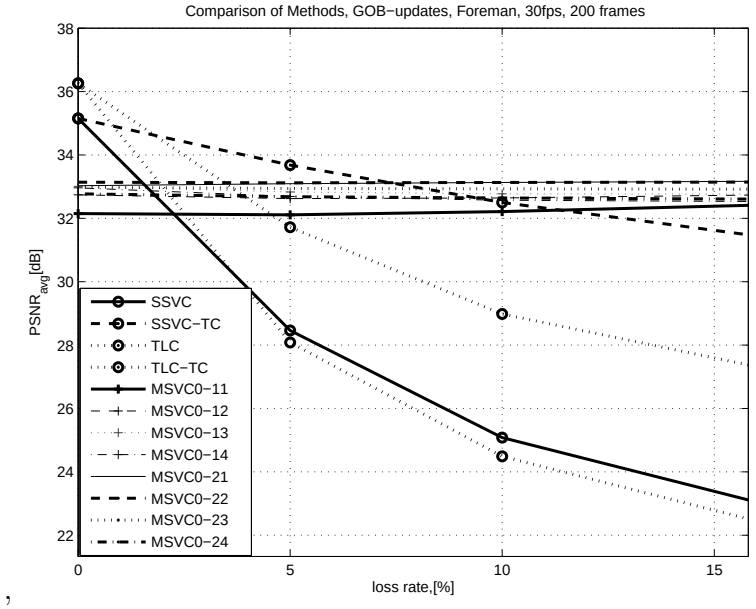
,

Figure 4.4: Comparison of SSVC, SSVC-TC, TLC, TLC-TC and MSVC, one of the MSVC channels is lossless, Akiyo.



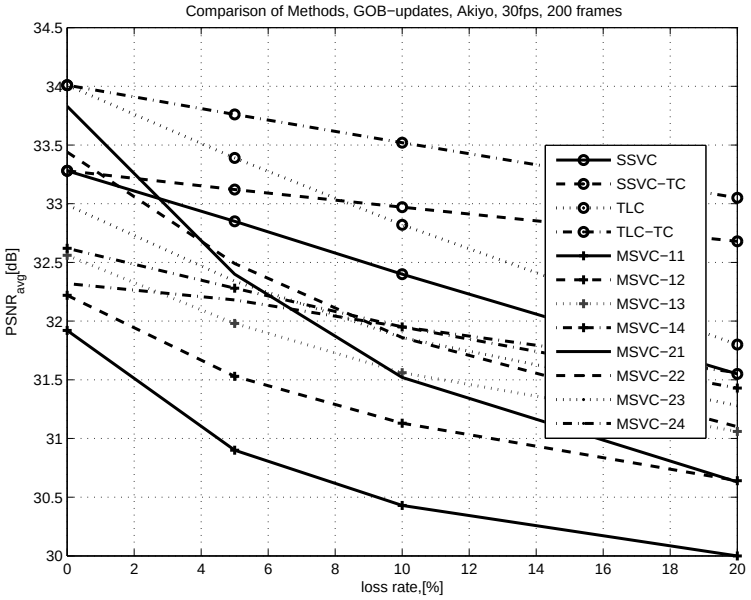
,

Figure 4.5: Comparison of SSVC, SSVC-TC, TLC, TLC-TC and MSVC, all channels have the same loss rate, Foreman with GOB-intra-updates



,

Figure 4.6: Comparison of SSVC, SSVC-TC, TLC, TLC-TC and MSVC, one of the MSVC channels is lossless, Foreman with GOB-intra-updates



,

Figure 4.7: Comparison of SSVC, SSVC-TC, TLC, TLC-TC and MSVC, all channels have the same loss rate, Akiyo with GOB-intra-updates

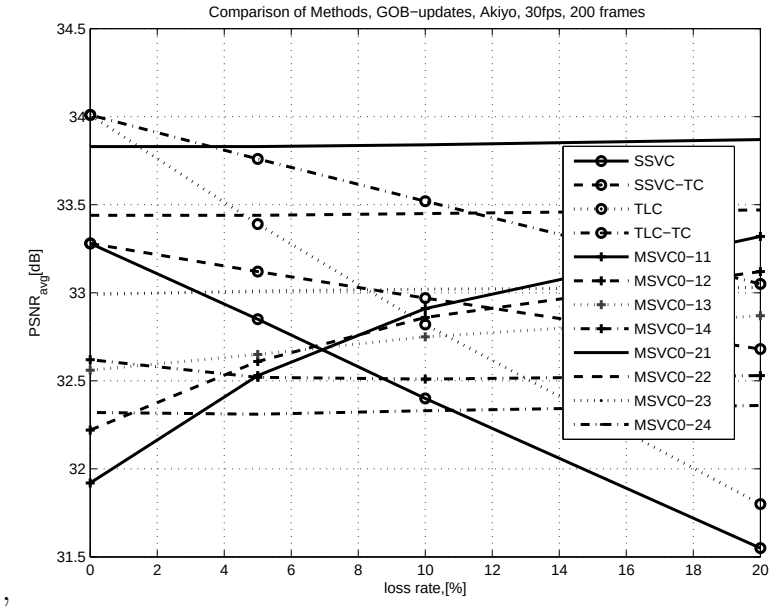


Figure 4.8: Comparison of SSVC, SSVC-TC, TLC, TLC-TC and MSVC, one of the MSVC channels is lossless, Akiyo with GOB-intra-updates

When frame-intra-updates are employed, the gap between methods becomes even smaller. At 20% the gap between SSVC-TC and MSVC is only 2 dB (Figure 4.9). Figure 4.10 shows the same for the lossless first channel case.

As mentioned previously, using frame-updates for Akiyo is no good idea, since the gain of motion compensation is very high. TLC seems to perform best, but this is since enhancement layer uses no updates, and this way bitrate is not wasted. $PSNR_{avg}$ is 0.5 dB larger than SSVC and 0.5 dB smaller than SSVC-TC at 20%, i.e. the differences are very small for Akiyo (Figure 4.11). MSVC outperforms also SSVC when the first channel is lossless (see Figure 4.12).

Figure 4.13 compares different coding modes for SSVC: 1- without intra-coding, 2- with GOB intra updates, 3- with frame-intra updates, 4- with ARQ(3), 5- with ARQ(3) and intra updates. The coding options for ARQ are given in Tables 4.1 and 4.2. Here we assumed at most 2 retransmissions if a packet is lost. The available bitrate for coding is then reduced to take the additional bitrate for retransmission into account. The reduction is smaller for 5% losses and respectively larger for 20% losses. The ARQ curves in this figure are actually constituted of 3 transmissions: for 5% transmission different quantization values are used as for 10% or 20% lossy

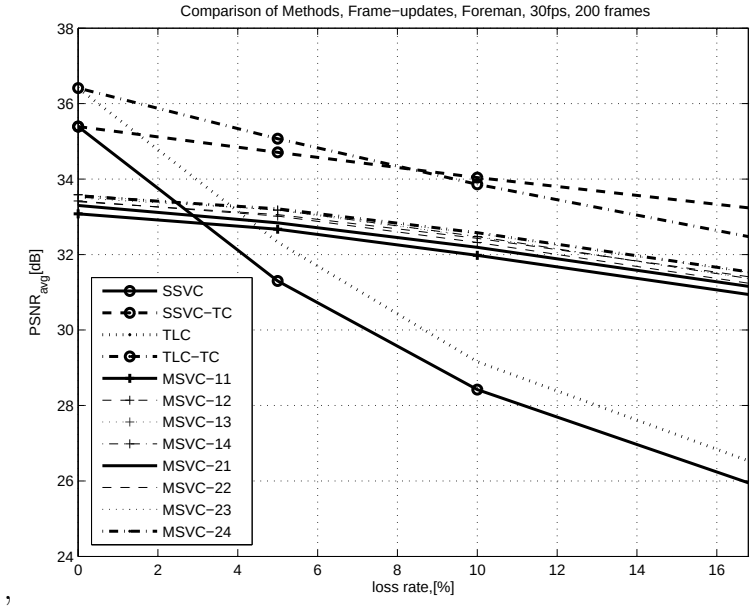


Figure 4.9: Comparison of SSVC, SSVC-TC, TLC, TLC-TC and MSVC, all channels have the same loss rate, Foreman with frame-intra-updates

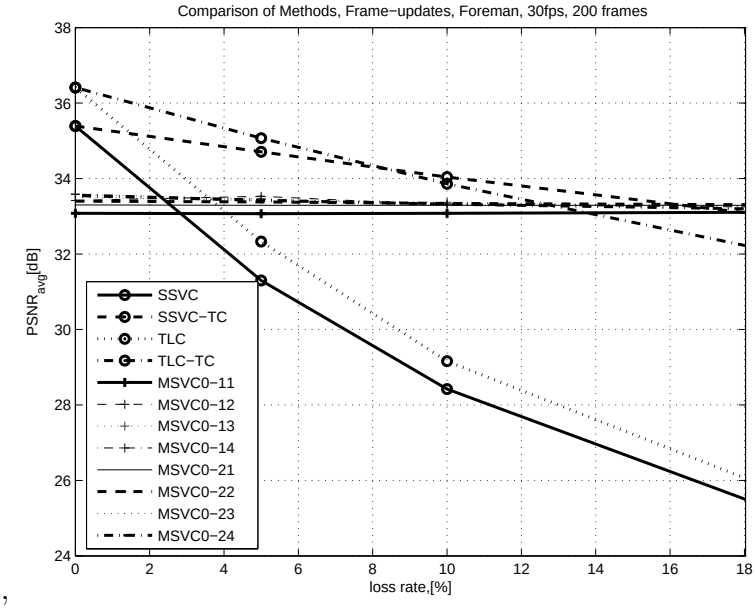
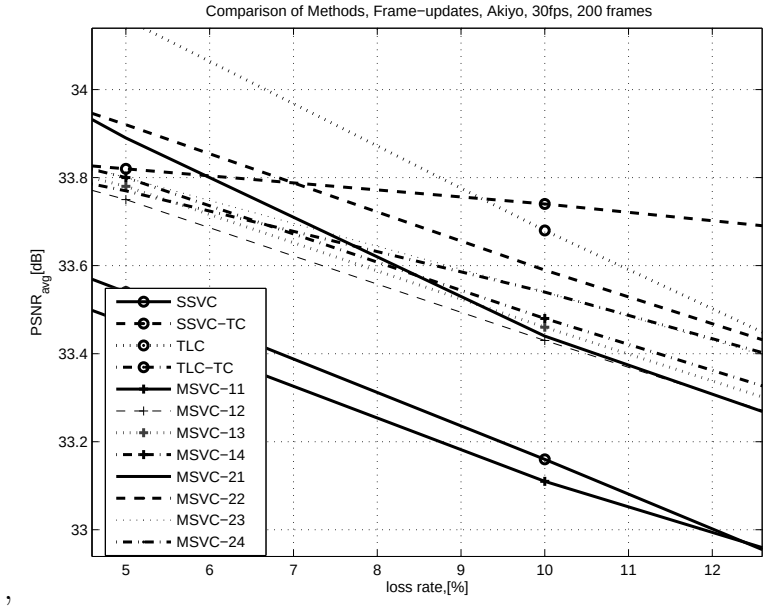
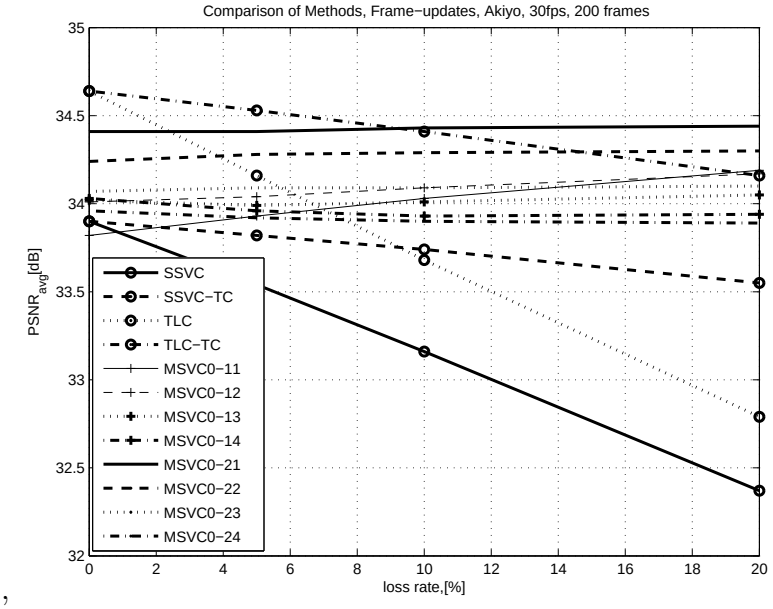


Figure 4.10: Comparison of SSVC, SSVC-TC, TLC, TLC-TC and MSVC, one of the MSVC channels is lossless, Foreman with frame-intra-updates



,

Figure 4.11: Comparison of SSVC, SSVC-TC, TLC, TLC-TC and MSVC, all channels have the same loss rate, Akiyo with frame-intra-updates



,

Figure 4.12: Comparison of SSVC, SSVC-TC, TLC, TLC-TC and MSVC, one of the MSVC channels is lossless, Akiyo with frame-intra-updates

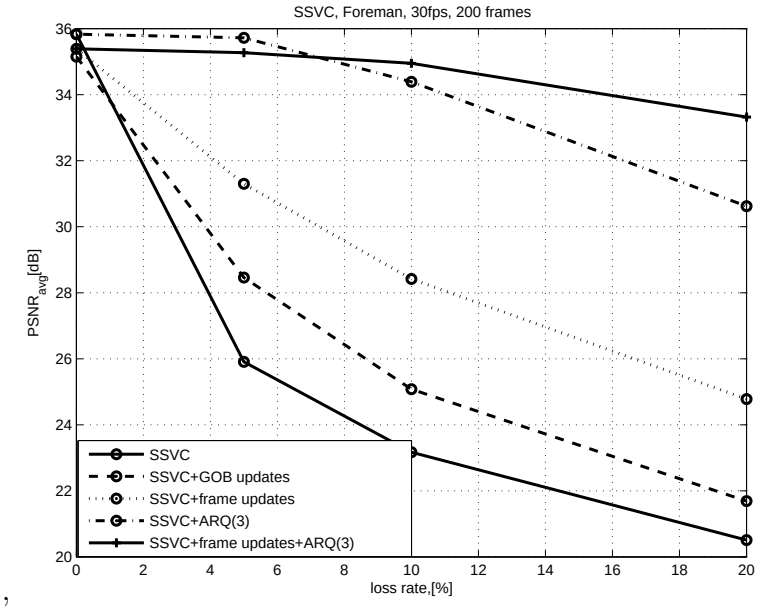


Figure 4.13: SSVC with different coding modes, Foreman

transmissions. The same applies for the combination of ARQ with frame-updates. $P_{t1} = 1 - p_{loss}$ is the probability of only one transmission for a packet, $P_{t2} = p_{loss}(1 - p_{loss})$ is the probability of two transmissions and $P_{t3} = p_{loss}^2$ is the probability of three transmissions. R_{avg} is then given as $R_{avg} = (P_{t1} + P_{t2} + P_{t3})R_{frame}$ where R_{frame} is the average bitrate required for the single transmission of a packet. Since R_{avg} is set for all coding methods, R_{frame} can be calculated from the above formula when p_{loss} is specified.

We can see from the figure that when retransmissions are allowed, ARQ(3) performs the best. The combination with frame updates is useful when loss rate increases. The delay caused by retransmissions and the bitrate for acknowledgements or negative acknowledgements are not taken into account which is necessary for a fair comparison. The frame-updates are followed by the GOB updates. In both cases incorporating updates pays off in a lossy environment. Using frame-updates is a more efficient method than GOB-updates since the state is refreshed completely at once.

The same comparison for Akiyo is shown in 4.14. Since intra-updates are not that effective for Akiyo, coding without intra update when possible with retransmissions yields the best results. Frame-updates make only sense when the loss rate increases beyond 16%. GOB updates are not efficient at all.

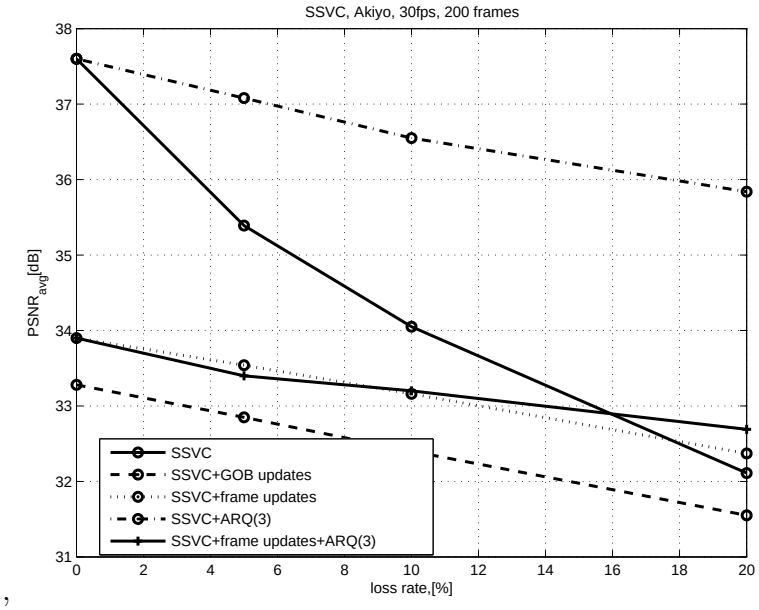


Figure 4.14: SSVC with different coding options, Akiyo

On the other hand, when for temporal concealment the real motion vectors are available as in case of SSVC-TC, ARQ performs the best followed by the frame updates at high loss rates. The same comparison is shown for Akiyo in 4.16. Here we see that there is actually no need for any special error protection, motion compensation solves the problem. ARQ becomes meaningful at high loss rates.

We get the results in Figure 4.17 for Foreman and in Figure 4.18 for Akiyo when the same kind of comparison is performed for TLC. The same comments as for SSVC applies also here.

Similar experiments are performed for TLC-TC. ARQ performs the best. Intra-frame-updates are followed by the no-intra-update transmission. GOB updates in the current configuration are fully useless since the updates concern only the base layer (see figure 4.19). The results for Akiyo are shown in Figure 4.20. The best performance is for ARQ and no-intra coding.

In the next experiment, MSVC coding modes are compared: Unbalanced, Approach 1 for Foreman in 4.21 and for Akiyo in 4.23. The same for balanced MSVC is presented for Foreman in 4.22 and for Akiyo in 4.24. Here we observe that when we can catch one lossless channel for a certain time whether we have balanced or unbalanced rate allocation, there is no need for any kind of intra-updates. State recovery

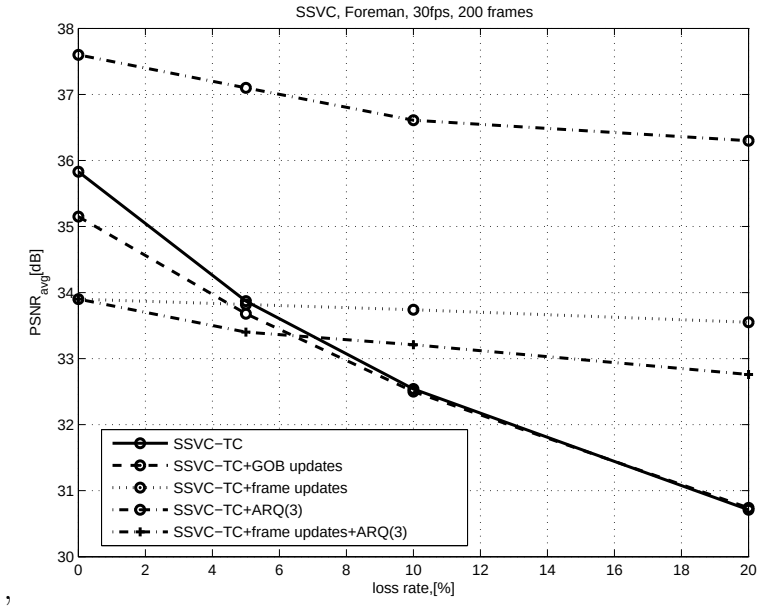


Figure 4.15: SSVC-TC with different coding options, Foreman

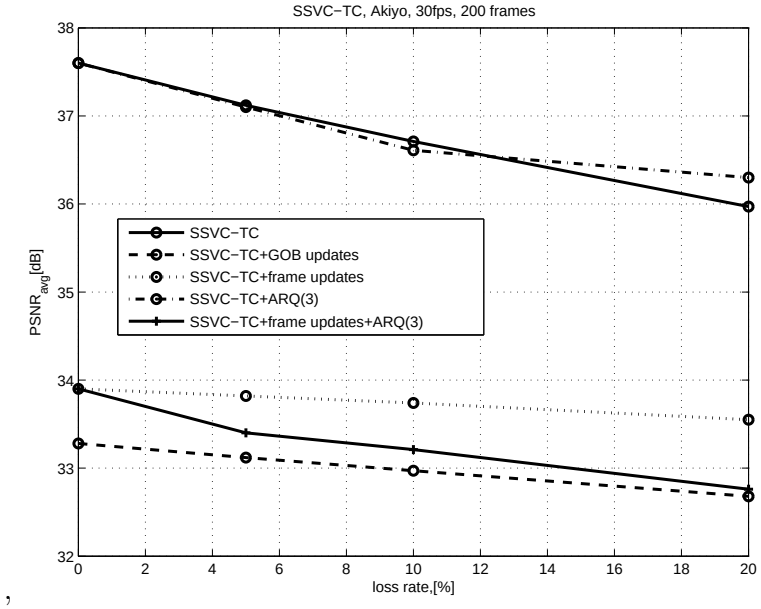


Figure 4.16: SSVC-TC with different coding options, Akiyo

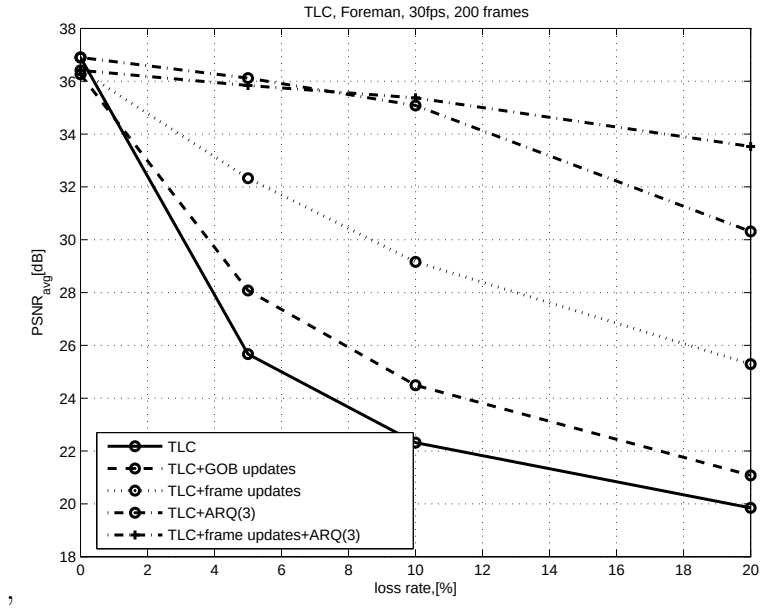


Figure 4.17: TLC with different coding options, Foreman

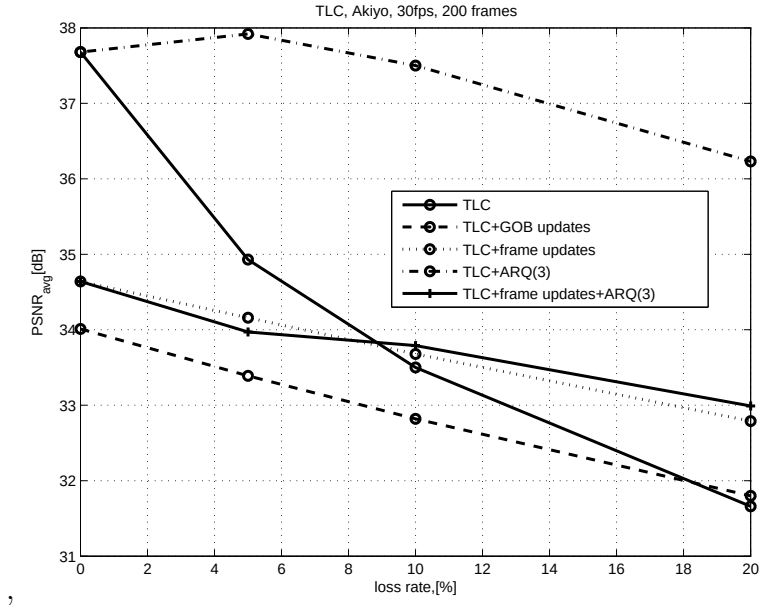


Figure 4.18: TLC with different coding options, Akiyo

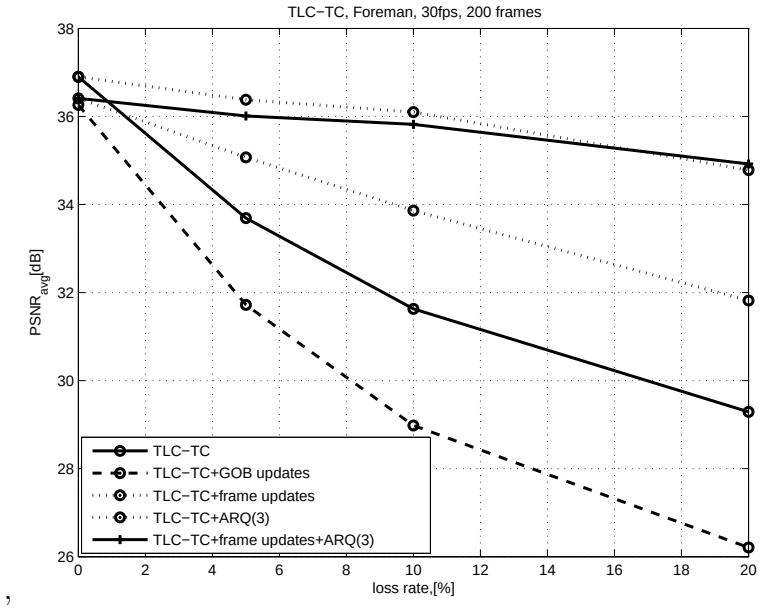


Figure 4.19: TLC-TC with different coding modes, Foreman

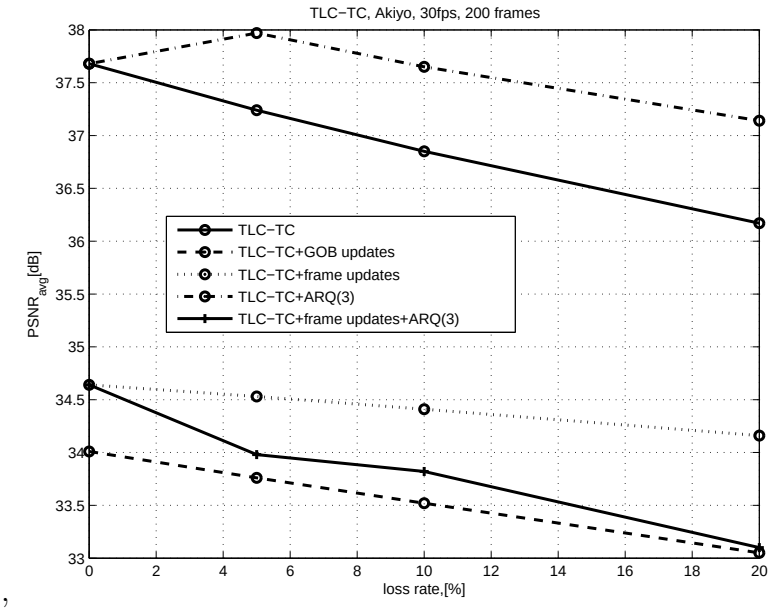
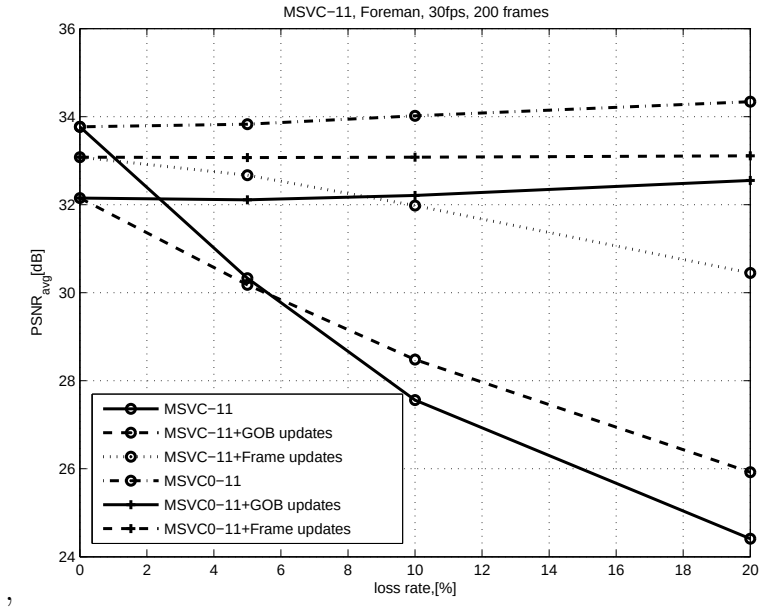


Figure 4.20: TLC-TC with different coding modes, Akiyo



,

Figure 4.21: Unbalanced MSVC with different coding modes, Foreman

works and keeps $PSNR_{avg}$ in a certain range. For Akiyo however, we observe again the disadvantage associated with intra updates.

If we perform the same experiment for Approach 2, the observation remains the same: $PSNR_{avg}$ changes, but the relation of the curves does not change.

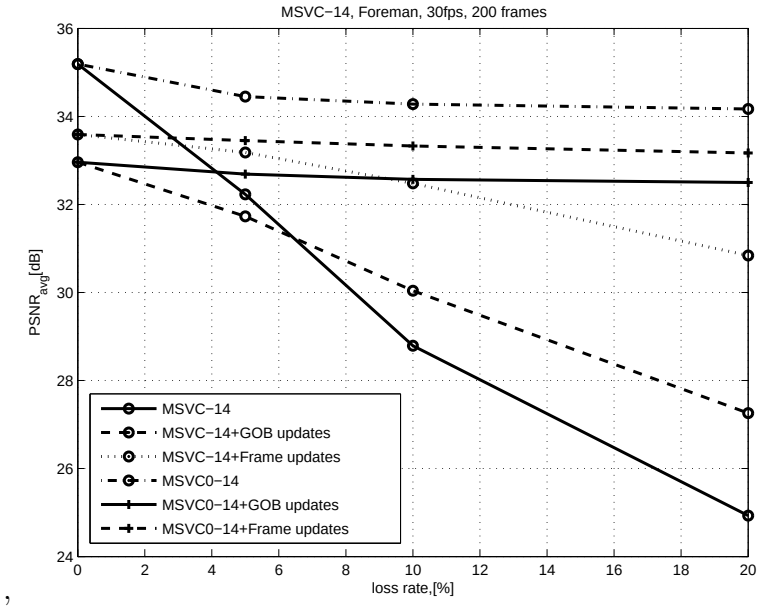


Figure 4.22: Balanced MSVC with different coding modes, Foreman

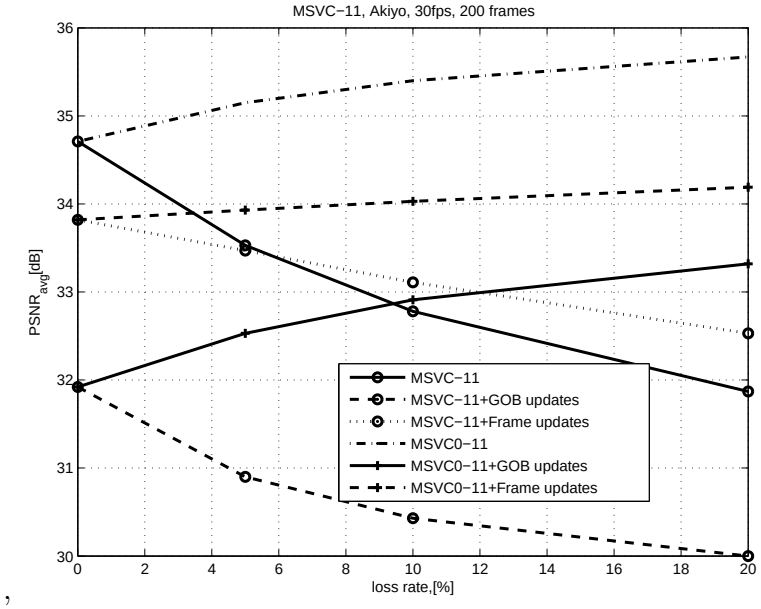
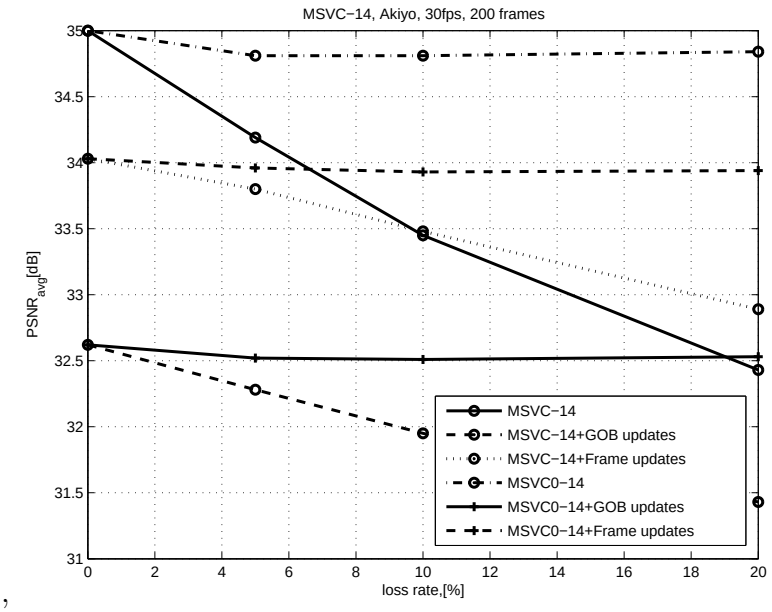


Figure 4.23: Unbalanced MSVC with different coding modes, Akiyo



,

Figure 4.24: Balanced MSVC with different coding modes, Akiyo

Chapter 5

Decoder Distortion Estimation for Single-State Coding

To determine the appropriate coding mode or even the coding and transmission technique, it is necessary to estimate the reconstructed signal quality depending on the channel conditions. Since channel conditions also vary, we assume that a method like channel probing is employed to track the loss rate or other related parameters (e.g. delay). If we can estimate the distortion on the reconstructed video sequence using the predicted channel loss rate, the decision about whether intra- or inter-coding, whether SSVC or MSVC, or the optimal rate allocation between several descriptions in MSVC is eased. But to easily adapt to changing channel conditions, an accurate but at the same time low complexity distortion estimation algorithm is required. In the sequel, a recursive estimation technique will be introduced based on AR(1) modelling of the video signal. The low implementation complexity, and high estimation accuracy of the proposed technique makes it particularly attractive for adaptive video communication applications, that try to optimize the streaming policy. The technique is developed first for Single-State Video Coding and extended to Temporal Layered Coding and Multi-State Video Coding in Chapters 6 and 7 respectively.

Hybrid coders employ inter-frame prediction to remove temporal redundancies and transform coding to remove spatial redundancies. Each video frame is divided into macroblocks which can be coded in intra or inter mode. In inter-mode, the macroblock (MB) is first predicted from the previously decoded frame via motion compensation then the prediction error is transform coded. In intra-mode, on the other hand, the

original MB data is directly transform coded. Intra coding is used as an important tool to reduce the effects of packet losses. To stop error propagation, the inter-frame prediction loop is switched off for certain blocks, so that the reproduced blocks are not dependent on past frames. However intra coding requires a higher bitrate than inter-coding. Clearly motion compensation leads to spatial propagation beyond MB boundaries since the pixels in the current MB may have been motion compensated from pixels in different MB's in the previous frame, each with a different error propagation history. In this work, we are interested in errors occurring due to packet losses. There are mainly two sources of packet losses in packet switched networks : buffer overflow at intermediate nodes of the network, and long queuing delays. The packet loss rate in internet communications may reach 20%. One application area for decoder distortion estimation is designing a rate distortion optimized mode selection which considers both the error concealment as well as quantization distortion. The loss probability can be deducted from the network conditions. Transcoding is another application area for distortion estimation.

The work presented in [169] computes the total decoder distortion recursively at pixel level precision to accurately account for spatial and temporal error propagation. The first and second moments of random variables (pixel luminance values) are needed, which increase the computation complexity. Moreover the computation is applicable to integer pixel accuracy becoming impractical for extension to half pixel accuracy or higher. Similarly the work in [133] introduces an analytical model to capture the effects of error concealment and interframe error propagation at the video decoder. The drawback of the system is the inadequate estimation accuracy of the overall distortion. Reference [115] gives a multidimensional bitrate control approach based on distortion measures.

Our approach is inspired by the work of Chen presented in [17] where he modeled motion-compensated hybrid coders and analyzed the characteristics of motion compensated frame differences. The video signal is modeled there as an AR(1) source. Another video modeling approach using hierarchical methods is given in [12]. Rate distortion estimation for video signals based on histograms is given [72]. The work presented in this chapter is partially described in [35].

5.1 AR(1) Process

The autoregressive (AR) process is usually used to model correlated signals. An N-th oder Markov process AR(N) is described by the following equation:

$$X(n) = Z(n) + \sum_{i=1}^{N} k_i X(n-i)$$

where k_i is the i^th weight. Based on this formula, the AR(1) process with zero mean and $k_1 = \rho$ is generated by:

$$\begin{aligned} X(n) &= Z(n) + \rho X(n-1) \\ &= Z(n) + \sum_{i=1}^{\infty} \rho^i Z(n-i) \end{aligned} \tag{5.1}$$

which is equivalent to the output of applying innovation signal Z(n) to a first-order all pole filter with frequency response as

$$H(e^{jw}) = \frac{1}{(1-\rho e^{-jw})}$$

The autocorrelation function $R_x(N)$ of X is given as:

$$R_x(N) = E[X(n)X(n+N)] \tag{5.2}$$

$$= \sigma_x^2 \rho^{|N|} \tag{5.3}$$

$$\rho_x(N) = R_x(N)/R_x(0)$$

where σ_x^2 is the variance of $X(n)$ and $\rho_x(N)$ is the variance normalized autocorrelation function.

The power spectrum density $S_x(e^{jw})$ is then calculated as:

$$S_x(e^{jw}) = \frac{1-\rho^2}{1+\rho^2-2\rho\cos w}\sigma_x^2$$

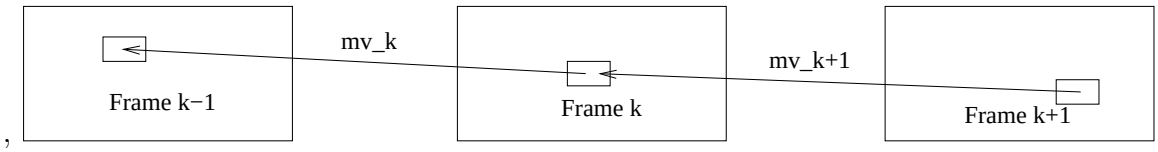


Figure 5.1: AR(1) source consisting of corresponding blocks along the sequence

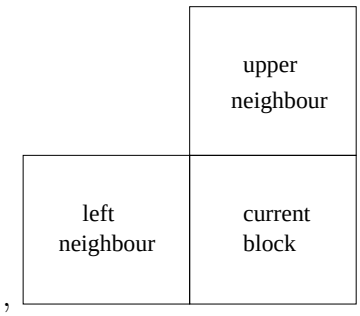


Figure 5.2: 4x4 Coding Blocks

5.2 Recursive Decoder Distortion Estimation

We assume that each frame whether I or P is transmitted in a separate and single packet. Each frame is divided into 4x4 pixel blocks and the decoder distortion estimate for each block of the frame is calculated depending on the packet loss parameter p_l. Block b in current frame f is supposed to constitute a sequence with its corresponding blocks on the other frames (determined using the motion vectors) along the video sequence (see Figure 5.1). We suppose that the error due to channel impairments and error concealment propagates only among the corresponding blocks along time. Although this assumption does not hold for sequences with occlusions and exposures and also for high motion sequences, the results achieved with the technique are quite successful as presented in section 5.3. Here we also assume indirectly that the corresponding blocks along time corresponds to an AR(1) source where a factor depending on the loop filtering corresponds to the correlation coefficient between blocks and the motion compensated block difference as well as the quantization distortion of the block constitute the additional noise term, which will be shown below.

For each frame f and for each block b given by coordinates (b_v, b_h) in f two distortion estimates are considered: $D_l(f, b_v, b_h)$ denoting the decoder distortion if f and this way b is lost whereas $D_r(f, b_v, b_h)$ denoting the decoder distortion if the

packet containing the data for f and so for b is received. b_v and b_h are the vertical and horizontal block coordinates respectively. $D(f, b_v, b_h)$ on the other hand, is the overall decoder distortion estimation calculated as the weighted average of $D_l(f, b_v, b_h)$ and $D_r(f, b_v, b_h)$:

$$D(f, b_v, b_h) = p_l D_l(f, b_v, b_h) + (1 - p_l) D_r(f, b_v, b_h)$$

We assume that the first frame is never lost. According to this, all blocks of the first frame have just the quantization distortion at the decoder:

$$\begin{aligned} D(1, b_v, b_h) &= D_r(1, b_v, b_h) \\ &= D_q(1, b_v, b_h) \end{aligned}$$

The blocks in the following frames can be intra- or inter-coded. If the block is intra-coded using intra-prediction modes we estimate its distortion using the average of the distortion estimates of its left and upper neighbors $D_{left}(f, b_v, b_h)$ and $D_{up}(f, b_v, b_h)$ respectively (see Figure 5.2):

$$D_{left}(f, b_v, b_h) = \begin{cases} 0 & \text{if } (b_h - 1) < 1 \\ D(f, b_v, b_h - 1) & \text{else} \end{cases} \tag{5.4}$$

and

$$D_{up}(f, b_v, b_h) = \begin{cases} 0 & \text{if } (b_v - 1) < 1 \\ D(f, b_v - 1, b_h) & \text{else} \end{cases} \tag{5.5}$$

resulting in

$$D_{old}(f, b_v, b_h) = \frac{D_{up}(f, b_v, b_h) + D_{left}(f, b_v, b_h)}{2}$$

$D_{old}(f, b_v, b_h)$ represents here distortion component coming from the reference

block. The component of $D_{old}(f, b_v, b_h)$ due to quantization is calculated as:

$$D_{old_q}(f, b_v, b_h) = \frac{D_{up_q}(f, b_v, b_h) + D_{left_q}(f, b_v, b_h)}{2}$$

.

On the other hand, If block b is inter-coded we have to consider the distortion of the corresponding block $b^{'}$ in the reference frame $f^{'} = f-1$ to calculate $D_{old}(f, b_v, b_h)$. $b^{'}$ has the coordinates $(b_v^{'}, b_h^{'})$ in $f^{'}$:

$$\begin{aligned} b_v^{'} &= b_v + round(m_v(f, b_v, b_h)/16) \\ b_h^{'} &= b_h + round(m_h(f, b_v, b_h)/16) \end{aligned}$$

where $m_v(f, b_v, b_h)$ and $m_h(f, b_v, b_h)$ are the vertical and horizontal components of the motion vector field at block position (b_h, b_v). The vector components are divided to 16, first, because we consider 4x4 pixel blocks and second, because of quarter pixel accuracy. If $b_v^{'}$ or $b_h^{'}$ lies out of the frame it is clipped into the frame.

For inter coded blocks we differentiate between $D_{old_l}(f, b_v, b_h)$ and $D_{old_r}(f, b_v, b_h)$, for the cases that the reference frame is lost and received respectively. $D_{old_l}(f, b_v, b_h)$, $D_{old_r}(f, b_v, b_h)$ and similarly $D_{old_q}(f, b_v, b_h)$, the quantization distortion of $b^{'}$ are given as:

$$\begin{aligned} D_{old_l}(f, b_v, b_h) &= D_l(f-1, b_v^{'}, b_h^{'}) \\ D_{old_r}(f, b_v, b_h) &= D_r(f-1, b_v^{'}, b_h^{'}) \\ D_{old_q}(f, b_v, b_h) &= D_q(f-1, b_v^{'}, b_h^{'}) \end{aligned}$$

The recursive formula to calculate the decoder distortion estimate $D(f, b_v, b_h)$ for block b has the recursion depth 1, i.e. we consider only the distortion of the corresponding block in the reference frame (the previous frame in the formulas) and we differentiate also between the cases whether this one is received or not. To incorporate the block intra updates into the recursive formula we differentiate also between intra-updated blocks and the rest. $D_{rr}(f, b_v, b_h)$ gives the distortion of block b if the previous

and the current frames are both received and $D_{ll}(f, b_v, b_h)$ gives the distortion if both are lost. Similarly $D_{lr}(f, b_v, b_h)$ gives the distortion when the previous one is lost but the current one is received , contrary $D_{rl}(f, b_v, b_h)$ the distortion when the previous one is received and the current one is lost.

If block b is intra-updated and also received, the decoder distortion is just the quantization distortion whether the previous frame is received or not:

$$\begin{aligned} D_{rr}(f, b_v, b_h) &= D_q(f, b_v, b_h) \\ D_{lr}(f, b_v, b_h) &= D_q(f, b_v, b_h) \end{aligned}$$

On the other hand, if b is received but not updated, D_{rr} and D_{lr} are given as:

$$\begin{aligned} D_{rr}(f, b_v, b_h) &= \alpha(D_{old_r}(f, b_v, b_h) - D_{old_q}(f, b_v, b_h)) \\ &\quad + D_q(f, b_v, b_h) \\ D_{lr}(f, b_v, b_h) &= \alpha(D_{old_l}(f, b_v, b_h) + D_{old_q}(f, b_v, b_h)) \\ &\quad + D_q(f, b_v, b_h) \end{aligned}$$

where α is a constant depending on the sequence, particularly on the scene activity representing the effect of the loop filter in hybrid coders. Note that block b is given as:

$$b = \sqrt{\alpha}(b^{'} + q^{'}) + mcbd(b)$$

where $b^{'}$ is the corresponding block of block b in the reference frame, $mcbd(b)$ is the motion compensated block difference and q and $q^{'}$ are the quantization distortions on blocks b and $b^{'}$ respectively. According to this formula, b can be considered as an AR(1) source where $\sqrt{\alpha}$ corresponds to the correlation coefficient and $\sqrt{\alpha}q^{'} + mcbd(b)$ is the additional noise term. The noise term is approximated as uncorrelated with b, which is only the case if motion compensation is ideal. Denoting the accumulated distortion on block $b^{'}$ as $d^{'}$ and on block b as d, we have:

$$\begin{aligned} b + d &= \sqrt{\alpha}(b^{'} + d^{'}) + mcbd(b) + q \\ d &= \sqrt{\alpha}(d^{'} - q^{'}) + q \end{aligned}$$

If the current and the previous frames are both received:

$$d^{'} = q^{'} + d_{rem}$$

where d_{rem} is the component of $d^{'}$ which is independent of $q^{'}$. The correlation between $d^{'}$ and $q^{'}$ is given as:

$$E[d^{'}q^{'}] = D_{old_q}$$

resulting in the formula for D_{rr}:

$$\begin{aligned} D_{rr}(f, b_v, b_h) &= E[d^2] \\ &= \alpha(D_{old_r}(f, b_v, b_h) + D_{old_q}(f, b_v, b_h) \\ &\quad -2D_{old_q}(f, b_v, b_h)) + D_q(f, b_v, b_h) \\ &= \alpha(D_{old_r}(f, b_v, b_h) - D_{old_q}(f, b_v, b_h)) \\ &\quad +D_q(f, b_v, b_h) \end{aligned}$$

On the other hand, if the previous frame is lost, $d^{'}$ and $q^{'}$ are uncorrelated to each other so that $D_{lr}(f, b_v, b_h)$ is calculated as:

$$\begin{aligned} D_{lr}(f, b_v, b_h) &= E[d^2] \\ &= \alpha(D_{old_r}(f, b_v, b_h) + D_{old_q}(f, b_v, b_h)) \\ &\quad +D_q(f, b_v, b_h) \end{aligned}$$

If the current frame is lost, the distortions $D_{rl}(f, b_v, b_h)$ and $D_{ll}(f, b_v, b_h)$ are calculated as follows:

$$\begin{aligned} D_{rl}(f, b_v, b_h) &= \alpha(D_{old_r}(f, b_v, b_h) - D_{old_q}(f, b_v, b_h)) \\ &\quad +pow\_mcbd(f, b_v, b_h) \end{aligned}$$

$$\begin{aligned} D_{ll}(f,b_v,b_h) &= \alpha(D_{old_l}(f,b_v,b_h) + D_{old_q}(f,b_v,b_h)) \\ &\quad + pow\_mcbd(f,b_v,b_h) \end{aligned}$$

where $pow\_mcbd(f,b_v,b_h)$ is the averaged sum of the squared pixel intensities of the motion compensated block difference.

Again considering $b^{'}$ when the current frame is lost:

$$\begin{aligned} b+d &= \sqrt{\alpha}(b^{'}+d^{'}) \\ d &= \sqrt{\alpha}(b^{'}+d^{'}) - \sqrt{\alpha}(b^{'}+q^{'}) - mcbd(b) \\ &= \sqrt{\alpha}(d^{'}-q^{'}) - mcbd(b) \end{aligned}$$

If the previous frame is received, $q^{'}$ and $d^{'}$ are correlated and otherwise uncorrelated to each other. The distortion is again calculated as $E[d^2]$ resulting in the equations given for $D_{rl}(f,b_v,b_h)$ and $D_{ll}(f,b_v,b_h)$ above.

$D_l(f,b_v,b_h)$ and $D_r(f,b_v,b_h)$, on the other hand, are calculated as the weighted averages of $D_{ll}(f,b_v,b_h)$ and $D_{rl}(f,b_v,b_h)$ and similarly of $D_{lr}(f,b_v,b_h)$ and $D_{rr}(f,b_v,b_h)$ respectively:

$$\begin{aligned} D_l(f,b_v,b_h) &= p_l D_{ll}(f,b_v,b_h) + (1-p_l)D_{rl}(f,b_v,b_h) \\ D_r(f,b_v,b_h) &= p_l D_{lr}(f,b_v,b_h) + (1-p_l)D_{rr}(f,b_v,b_h) \end{aligned}$$

The estimated PSNR for each frame, PSNR(f) is calculated as:

$$\begin{aligned} PSNR(f) &= {p_l}^2 PSNR_{ll}(f) + p_l(1-p_l)PSNR_{lr}(f) \\ &\quad + (1-p_l)^2 PSNR_{rr}(f) \\ &\quad + (1-p_l)p_l PSNR_{rl}(f) \end{aligned}$$

where the individual PSNR values are given as:

$$PSNR_{ll} = 10log_{10}\left(\frac{255^2}{\frac{\sum_{i=1}^{max_i}\sum_{j=1}^{max_j} D_{ll}(f,i,j)}{total\_block\_number}}\right)$$

and *total_block_number* is the total number of blocks in a frame.

The remaining PSNR values are calculated by exchanging the subscripts of the variables respectively.

5.3 Experiments and Results

We assume that each frame whether I or P frame is transmitted in a separate and single packet. The packets are lost with a given uniform loss rate of p_l. If a frame is lost assuming that the exact motion information is known, we use motion compensated error concealment, i.e. the motion compensated difference value for these blocks is set to zero. The lossy channel with a random uniform loss generator and sequences are coded and decoded with H. 264 Codec (TML Version 9.0). For each loss rate 100 different loss patterns are considered. We used five QCIF video sequences coded at 30 fps: Foreman, Akiyo, Claire, Mother & Daughter and Salesman for the simulations and the estimations. We considered three cases for each sequence: 1-without any intra-updates, 2-with updates of GOB's, 3-with intra-frame updates. To generate the estimation values we needed four parameters: 1- quantization distortion, 2-motion vector field, 3-energy of motion compensated block differences (*pow_mcbd*) for each block of each frame and 4-α. All of these parameters are calculated off-line at the sender side. Quantization distortion should be adapted to the coding mode to simplify the calculations.

5.3.1 without intra-updates

Tables 5.1 and 5.2 list the coding parameters for the five sequences when non-intra coding was used. Table 5.1 lists the quantization step sizes which are common for all coding modes whereas Table 5.2 gives the bitrate and coded frame PSNR for no-intra-coding. Similar tables are given for intra-GOB-coding and for intra-frame-coding in their corresponding subsections.

Figure 5.3 depicts frame PSNR over frame number and shows the accordance of the model results with the simulation results for the Foreman sequence at different loss rates. As seen the prediction errors are within 0.5 dB when the first 100 frames of the sequence are considered. As the loss rate increases prediction accuracy decreases. Another point is that curves generated by the model provides smooth transitions

	QP first	QP remain.
F.	13	14
A.	18	18
C.	18	18
M&D.	13	14
S.	13	14

Table 5.1: Quantization Parameters

	avg. PSNR [dB] no-i.-updates	Bitrate [kbits/s] no-i.-updates	i.-GOB period	i.-frame period
F.	37.14	189.13	0	0
A.	37.10	18.17	0	0
C.	38.53	22.42	0	0
M&D.	37.96	86.12	0	0
S.	37.51	68.44	0	0

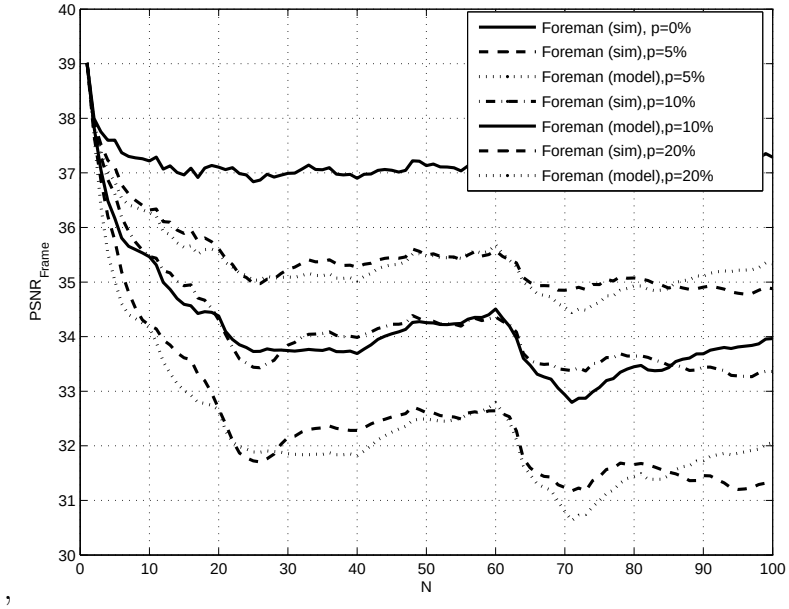
Table 5.2: no-intra-coding Parameters

between neighboring frame PSNR's, therefore the simulation curve cannot be tracked precisely by the model curve. Figure 5.4 shows the similar plot for the Akiyo sequence. The frame PSNR differences between reconstructions corresponding to different loss rates are smaller since error concealment works very good. The differences between simulation and model results are also confined within 0.25 dB. Because of the very small differences between results of different loss rates, the accordance between the model and simulation results becomes visible first after the 40^{th} frame.

Similarly the model accordance for Claire is shown in Figure 5.5. The accuracy is within the 0.5 dB region.

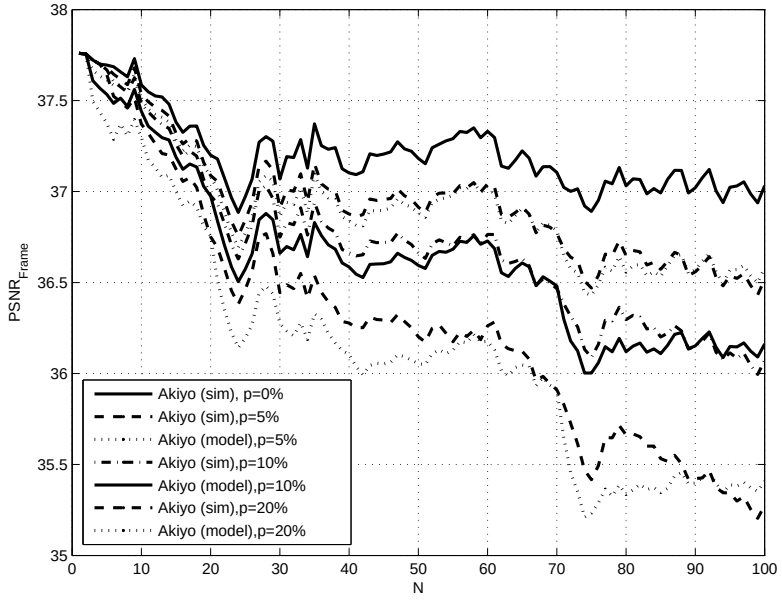
Mother & Daughter and Salesman are both relatively high motion sequences as compared to Akiyo and Claire. The prediction errors are still smaller than 0.5 dB for the first 100 frames. Results are shown in Figures 5.6 and 5.7.

Figure 5.8 shows the course of α for all sequences over the frame number. We let α to adapt to the sequence in blocks of 10 frames. Akiyo and Claire have the smallest α values whereas Foreman, Mother & Daughter and Salesman have the largest. Larger α values point to higher spatial or temporal motion. Moreover as the segment number increases α increases also. The largest increment in α occurs after the first time segment. The increase in α is mainly due to the accumulation of the



,

Figure 5.3: PSNR over Frame Number, Model and Simulation Accordance, Foreman, no intra updates



,

Figure 5.4: PSNR over Frame Number, Model and Simulation Accordance, Akiyo, no intra updates

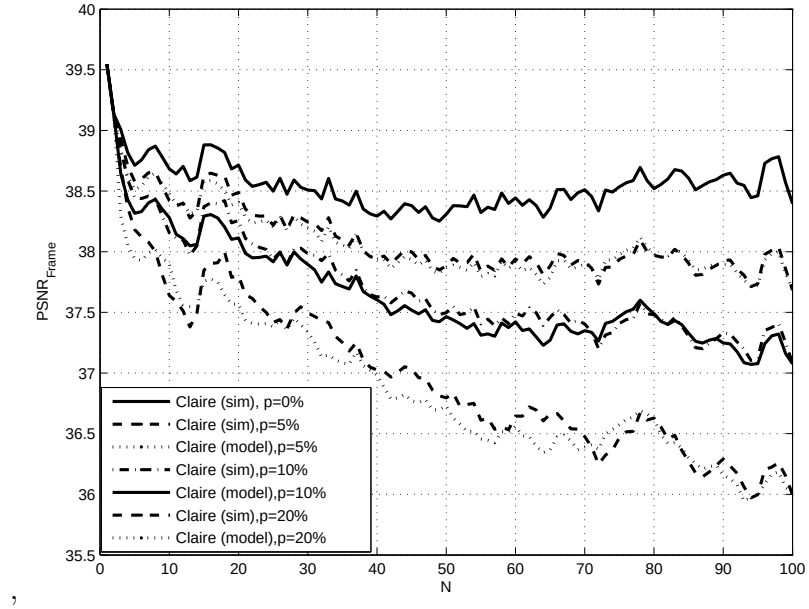


Figure 5.5: PSNR over Frame Number, Model-Simulation Accordance, Claire, no intra updates

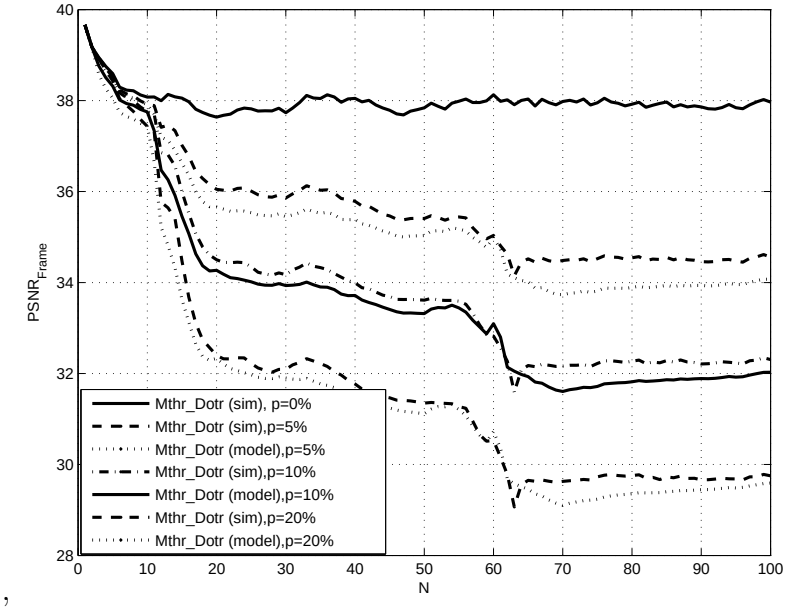
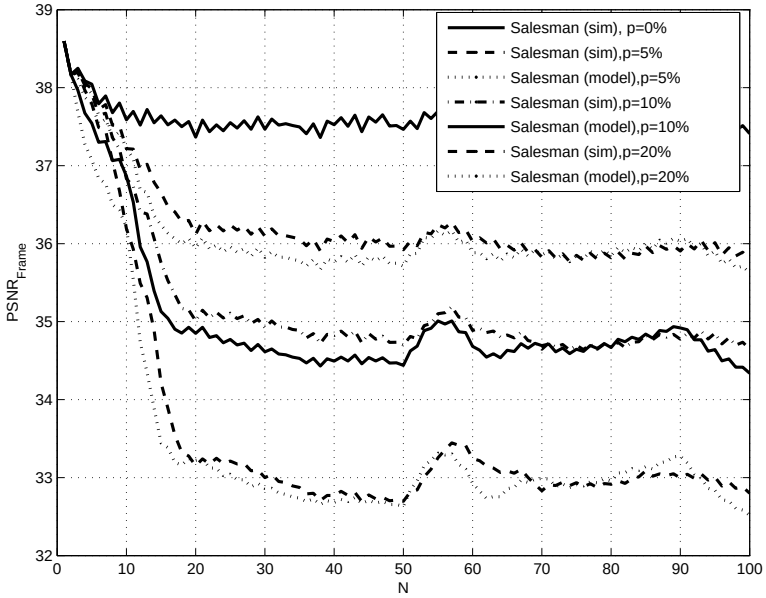


Figure 5.6: PSNR over Frame Number, Model-Simulation Accordance, Mother & Daughter, no intra updates

]



,

Figure 5.7: PSNR over Frame Number, Model-Simulation Accordance, Salesman, no intra updates

estimation errors over time. In the first segment there is no accumulation yet and therefore the α value is the smallest.

5.3.2 with intra-GOB-updates

Next we investigate the model-simulation accordance when GOB updates are considered. Table 5.3 shows the intra-update parameters as well as the coding rate and the encoded frame PSNR's of the sequences.

The results for the sequences Foreman, Akiyo, Claire, Mother & Daughter and

	avg. PSNR [dB] i.-GOB-updates	**Bitrate [kbits/s] i.-GOB-updates**	**i.-GOB period**	**i.-frame period**
F.	37.05	264.59	1	0
A.	36.75	72.56	1	0
C.	38.60	58.96	1	0
M&D.	37.79	159.41	1	0
S.	36.58	175.20	1	0

Table 5.3: intra-GOB-coding Parameters

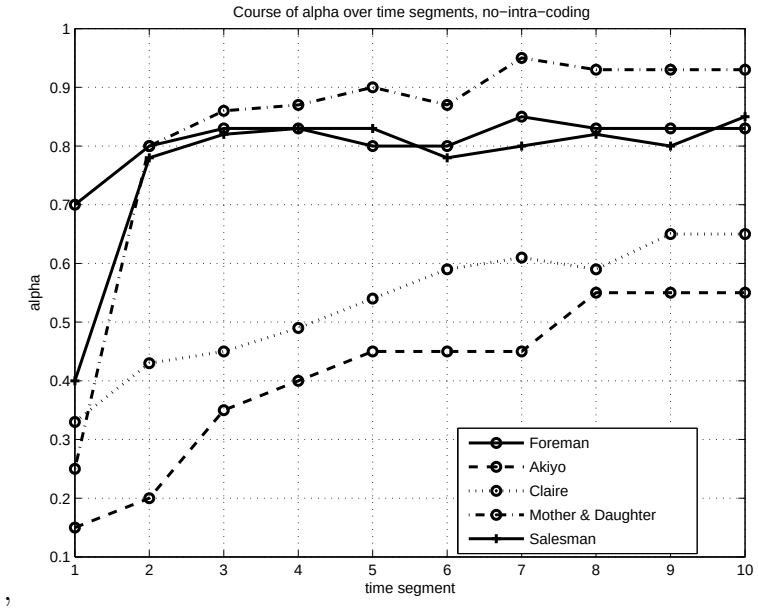


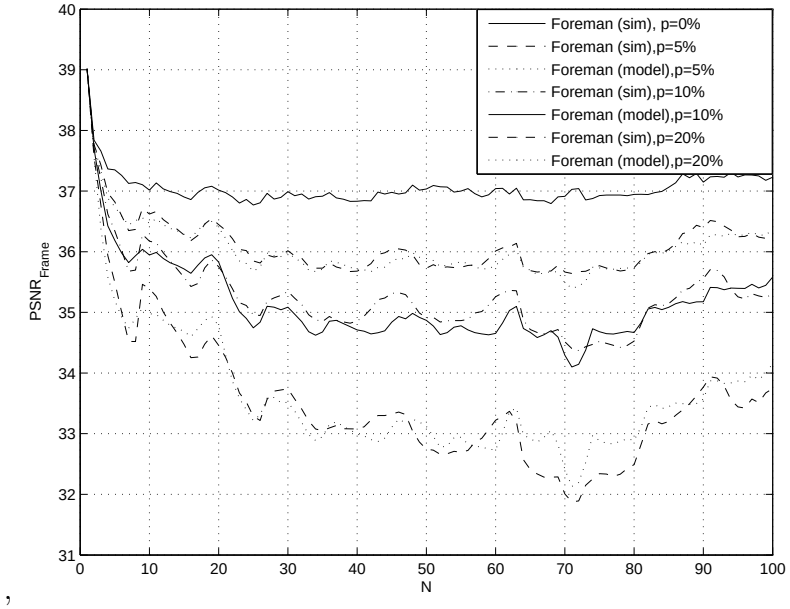
Figure 5.8: Course of α over time segments, no-intra-coding

Salesman are shown in Figures 5.9, 5.10, 5.11, 5.12 and 5.13. Estimation accuracy is only decreased for the Mother & Daughter sequence. The periodically updated GOB's are modeled as intra coded blocks at the corresponding block coordinates.

Course of α is depicted over time segments, each consisting of 10 frames. Comparing 5.14 with Figure 5.8 we see that the α values are slightly increased. But there is no steady increase, the value stabilizes after some time and even decreases slightly.

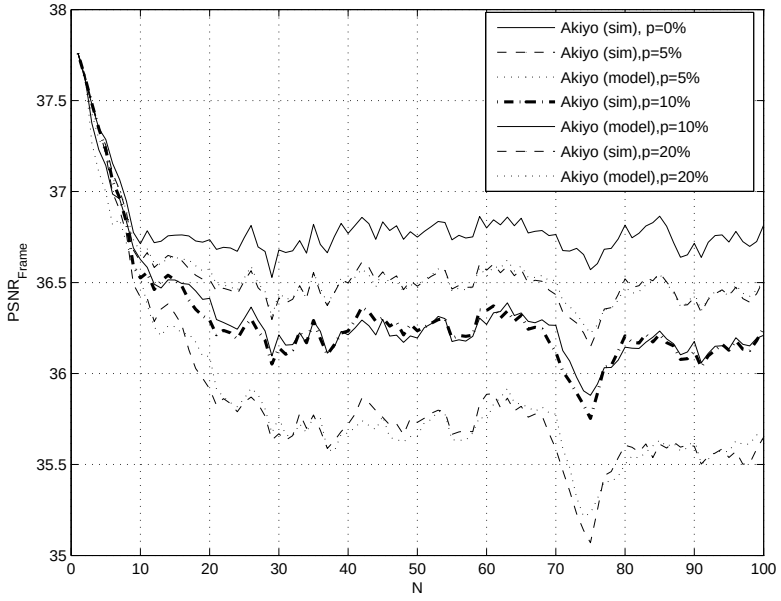
5.3.3 intra-frame-updates

The model-simulation accordance in presence of intra-frame-updates is investigated next. Table 5.4 shows the intra-update coding parameters as well as the coding rate and the encoded frame PSNR's for the sequences. Figures 5.15, 5.16, 5.17, 5.18 and 5.19 show the intra-frame-update incorporated results for Foreman, Akiyo, Claire, Mother & Daughter and Salesman respectively. The peaks in the figures are due to the big difference of quantization levels of the intra and inter-coded frames. Smaller differences in quantization levels ease the prediction of the simulation curve. The difference is set extra big, to test the model capabilities. Since the model smoothes the transitions in its prediction, the estimation accuracy for the frames right after the intra-updated-frame is especially low. But the results are still very good and



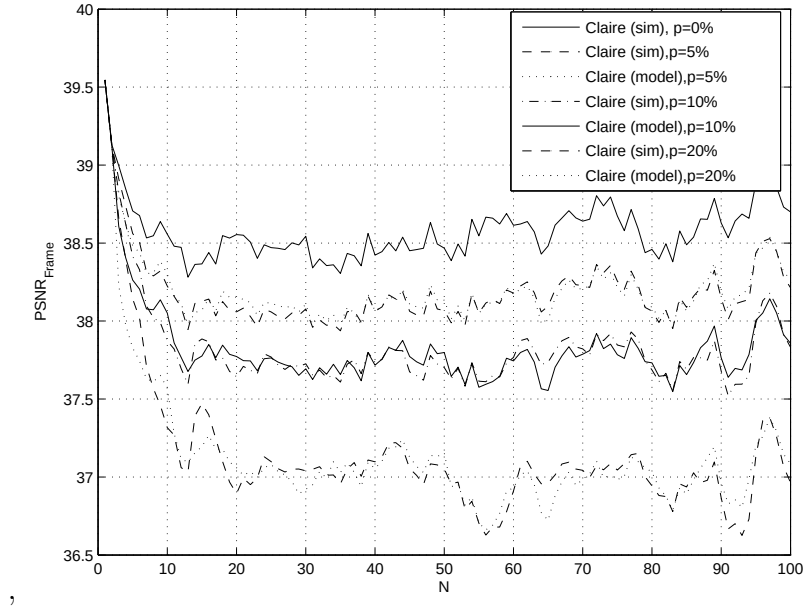
,

Figure 5.9: PSNR over Frame Number, Model-Simulation Accordance, Foreman, intra-GOB-updates



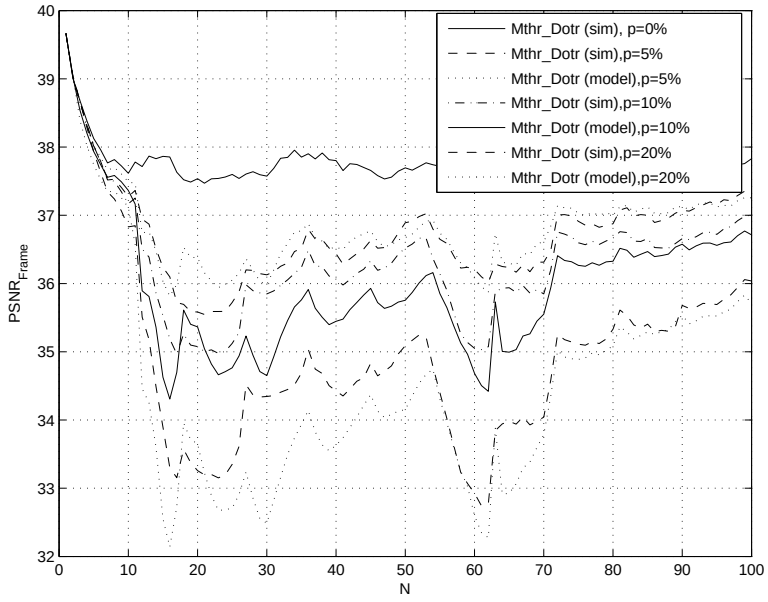
,

Figure 5.10: PSNR over Frame Number, Model Simulation Accordance, Akiyo, intra-GOB-updates



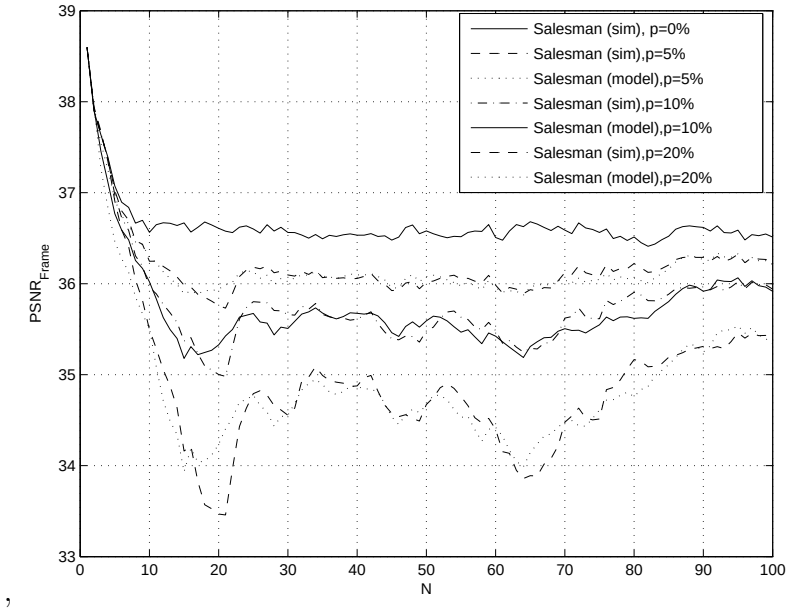
,

Figure 5.11: PSNR over Frame Number, Model-Simulation Accordance, Claire, intra-GOB-updates



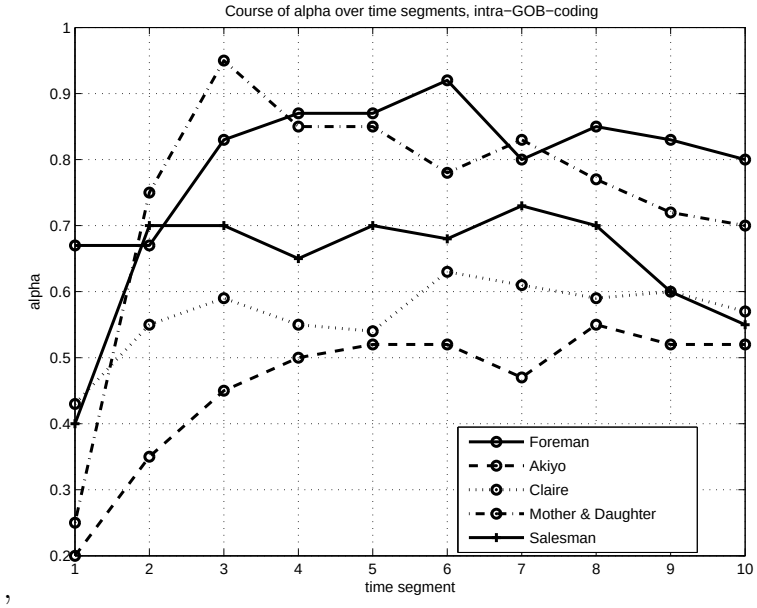
,

Figure 5.12: PSNR over Frame Number, Model-Simulation Accordance, Mother & Daughter, intra-GOB-updates



,

Figure 5.13: PSNR over Frame Number, Model-Simulation Accordance, Salesman, intra-GOB-updates

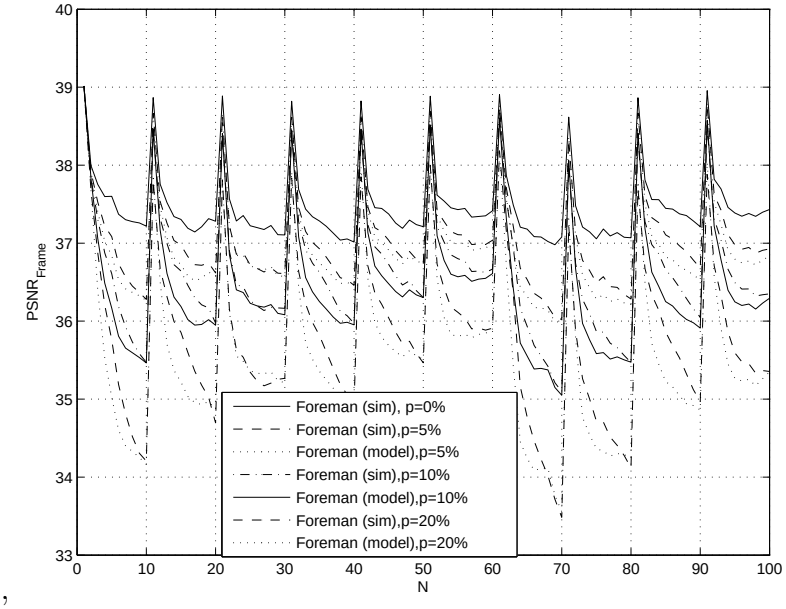


,

Figure 5.14: Course of α over time segments, intra-GOB-updates

	avg. PSNR [dB] i.-frame-updates	Bitrate [kbits/s] i.-frame-updates	i.-GOB period	i.-frame period
F.	37.53	261.76	0	10
A.	37.58	65.57	0	10
C.	39.10	54.20	0	10
M&D.	38.66	156.84	0	10
S.	37.98	174.74	0	10

Table 5.4: intra-frame-coding Parameters



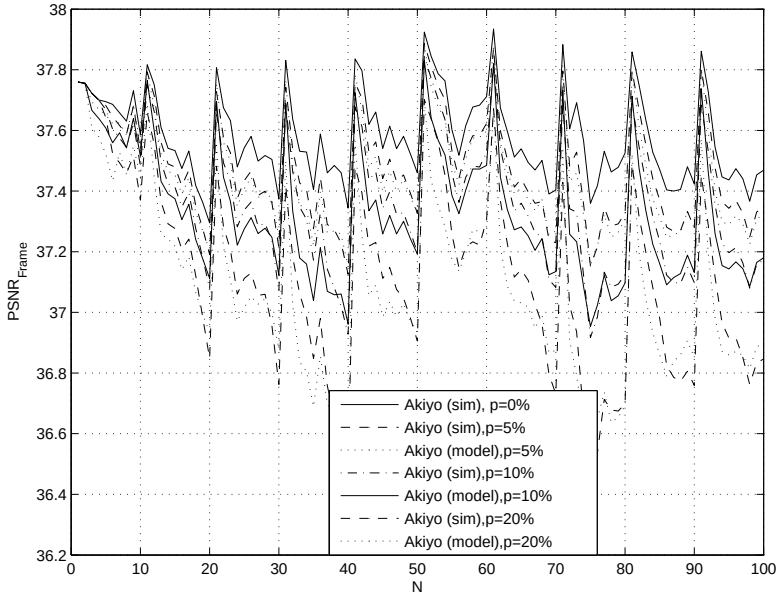
,

Figure 5.15: PSNR over Frame Number, Model-Simulation Accordance, Foreman, intra-frame-updates

estimation accuracy is confined within the 1 dB region.

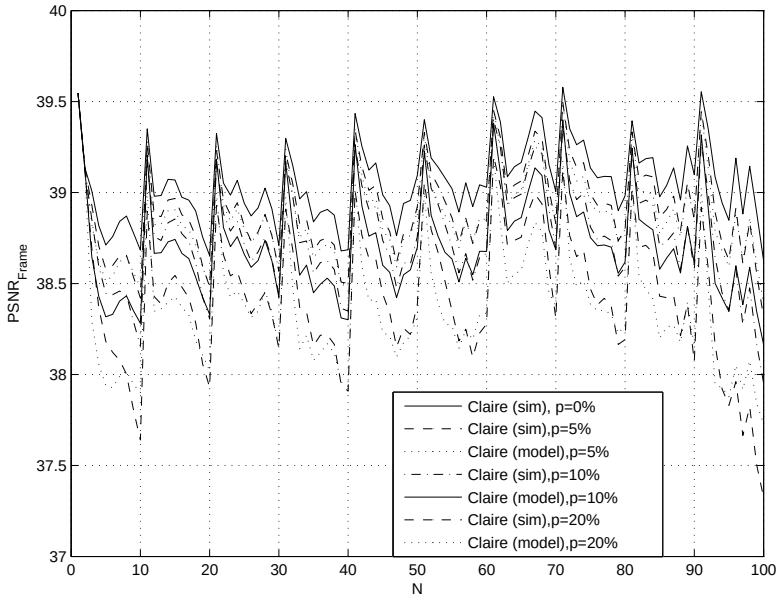
Course of α over time segments for intra-frame-updated sequences are given in 5.20, 5.21, 5.22, 5.23 and 5.24 for Foreman, Akiyo, Claire, Mother & Daughter and Salesman respectively. Since there is no estimation error accumulation due to intra-frame-updates, the variation in α is only dependent on the sequence in this time segment. The segments with high α values indicate high activity. But still Akiyo and Claire have the lowest α values, confirming the fact that they are low motion sequences.

Figures 5.21, 5.22, 5.23, 5.24 and 5.25 compare the α values over time segments for Foreman, Akiyo, Claire, Mother & Daughter and Salesman respectively at different



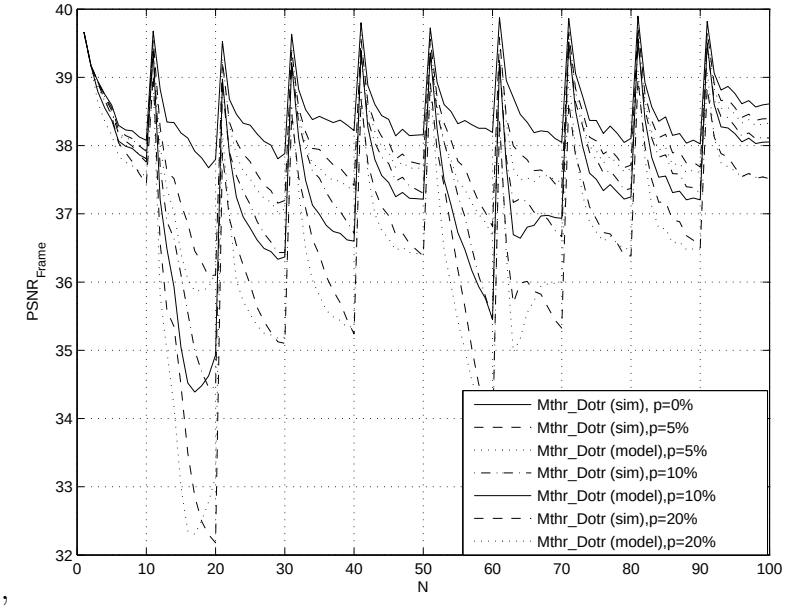
,

Figure 5.16: PSNR over Frame Number, Model-Simulation Accordance, Akiyo, intra-frame-updates



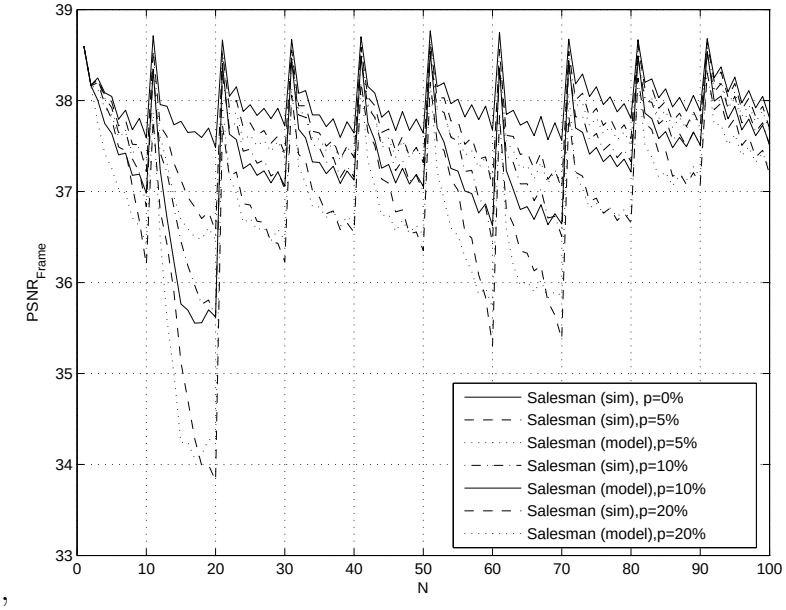
,

Figure 5.17: PSNR over Frame Number, Model-Simulation Accordance, Claire, intra-frame-updates



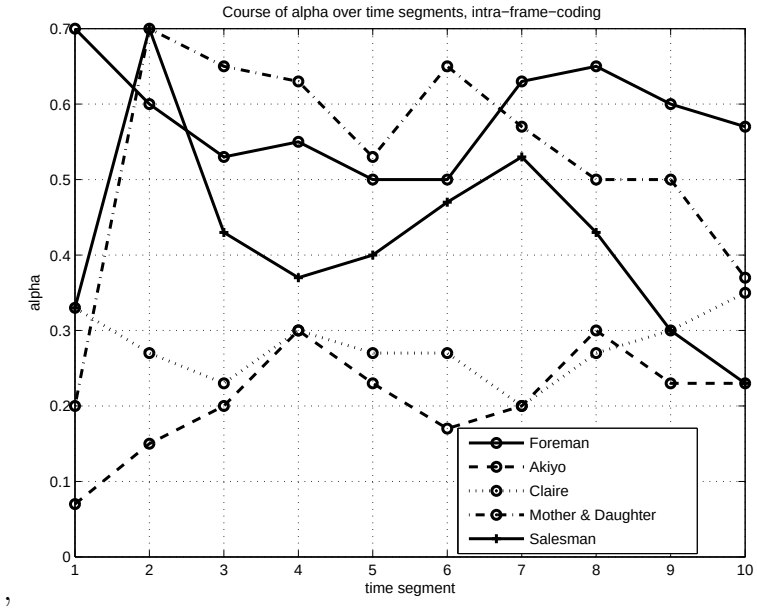
,

Figure 5.18: PSNR over Frame Number, Model-Simulation Accordance, Mother & Daughter, intra-frame-updates



,

Figure 5.19: PSNR over Frame Number, Model-Simulation Accordance, Salesman, intra-frame-updates



,

Figure 5.20: Course of α over time segments, intra-frame-updates

coding options, i.e. with or without intra coding. For all of the sequences, intra-frame updates require the smallest α values. For Foreman, Akiyo and Claire, GOB-intra-updates require the largest α values. Mother & Daughter and Salesman have the largest values for no-intra-coding.

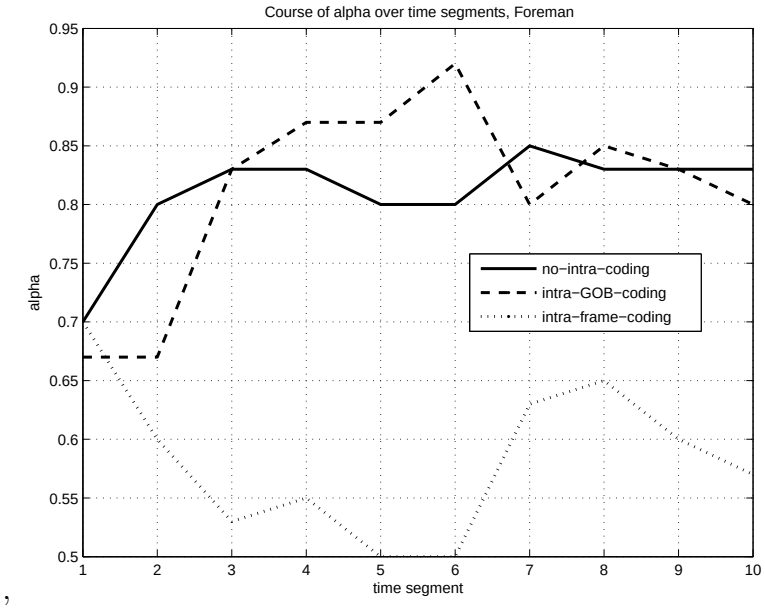


Figure 5.21: Course of α over time segments, Foreman

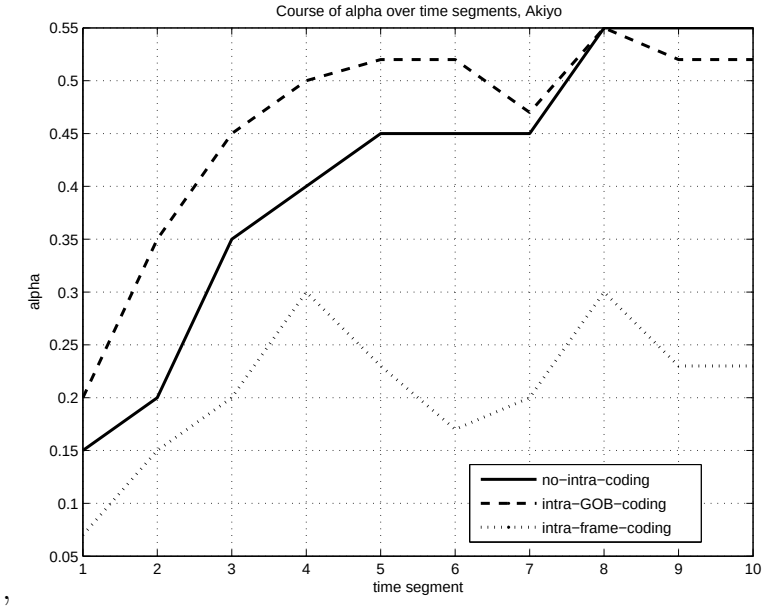
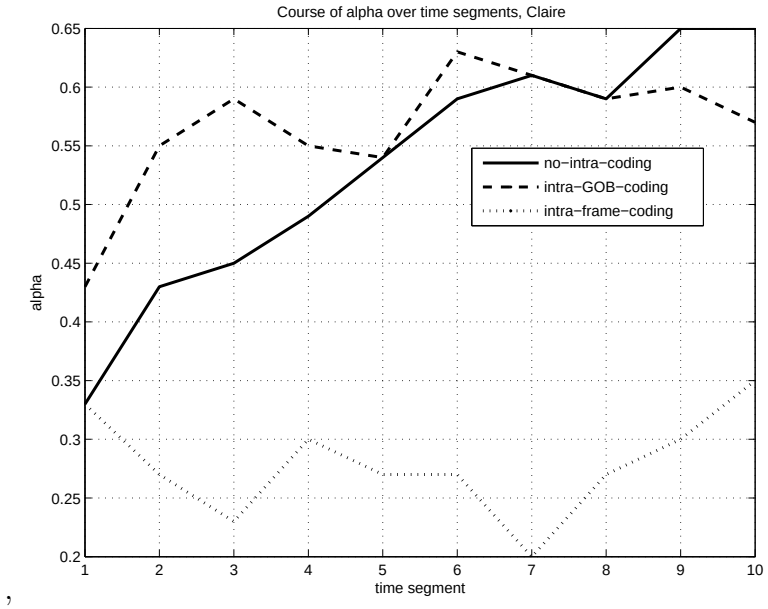
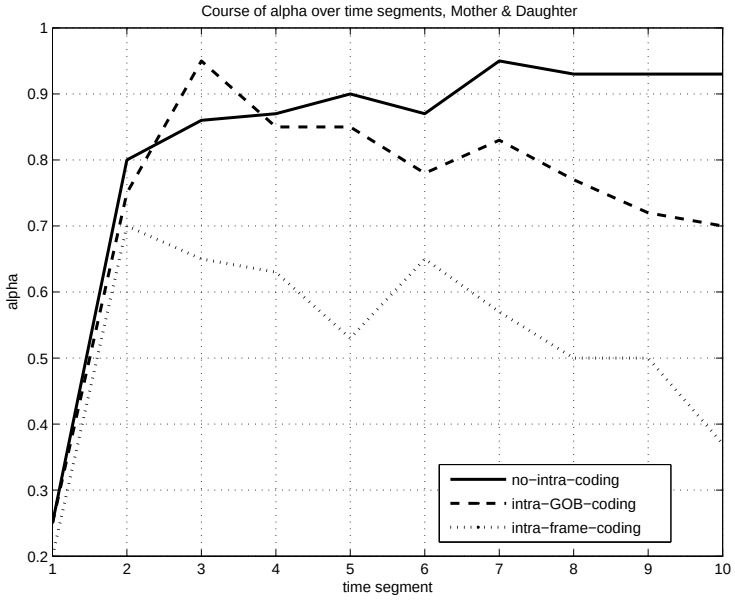


Figure 5.22: Course of α over time segments, Akiyo



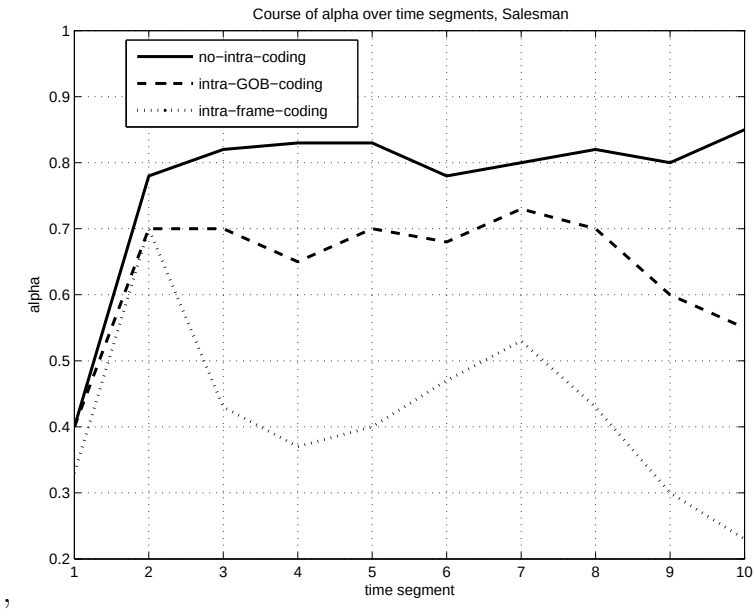
,

Figure 5.23: Course of α over time segments, Claire



,

Figure 5.24: Course of α over time segments, Mother & Daughter



,

Figure 5.25: Course of α over time segments, Salesman

Chapter 6

Decoder Distortion Estimation for Temporal Layered Coding

The AR(1) based model for decoder distortion estimation can be extended to layered coding also. In this section, we are interested in the decoder distortion estimation in a temporal layered transmission system. The results are depicted for the case that both the base and the enhancement layer undergo a lossy transmission with the same loss rate but independent losses. There is no need for a change in the model if separate layers are affected with different loss rates. Even the α does not change. Model correspondence is given for no-intra, intra-GOB-coding and intra-frame-coding in sections 6.1.1, 6.1.2 and 6.1.3 respectively.

6.1 Recursive Decoder Distortion Estimation: Extended to B-Frames

We again assume that each frame is transmitted in a single and separate packet, i.e. I, P or B-frames having different sizes each are transmitted in individual packets. In this section we will concentrate only on the differences due to the inclusion of the temporal enhancement layer. The details of the algorithm common with section 5.2 are omitted here.

For B-frames we not only have the forward motion vectors but also the backward motion vectors, i.e. $m_{f_h}(f, b_v, b_h)$ and $m_{f_v}(f, b_v, b_h)$ are the horizontal and vertical

components of the forward motion vector at frame f and block b with the horizontal and vertical coordinates b_h and b_v respectively. Similarly $m_{b_h}(f, b_v, b_h)$ and $m_{b_v}(f, b_v, b_h)$ are the horizontal and vertical components of the backward motion vector at frame f and block b with the horizontal and vertical coordinates b_h and b_v respectively. For P frames the backward motion vector field consists of zeros so that only the forward motion field is meaningful and to be used. Moreover the temporal distance between the reference and the predicted frame is doubled resulting in larger values of the forward motion vectors. Motion compensated difference frames *mcdf* are calculated as the difference between the current frame and the weighted average of the motion compensated forward and backward reference images. Let F be the current frame and F_{-1} and F_{+1} are the previous and the next frames in the sequence. Supposing that F is to be coded as a B-frame $mcdb(f, b_v, b_h)$, the motion compensated difference block with block coordinates (b_v, b_h) in f can be given as follows:

$$\begin{aligned} mcbd(f, b_v, b_h) &= F(b_v, b_h) - \frac{F_{+1}(b''_v, b''_h) + F_{-1}(b'_v, b'_h)}{2} \\ b'_v &= b_v + mv_{f_v}(f, b_v, b_h); \\ b'_h &= b_h + mv_{f_h}(f, b_v, b_h); \\ b''_v &= b_v + mv_{b_v}(f, b_v, b_h); \\ b''_h &= b_h + mv_{b_h}(f, b_v, b_h); \end{aligned}$$

For P-frames on the other hand we have the following relationships:

$$\begin{aligned} mcbd(f, b_v, b_h) &= F(b_v, b_h) - F_{-1}(b'_v, b'_h) \\ b'_v &= b_v + mv_{f_v}(f, b_v, b_h); \\ b'_h &= b_h + mv_{f_h}(f, b_v, b_h); \end{aligned}$$

For P frames, the step given in section 5.2 applies with the difference that the reference frame is the second preceding one with index $f - 2$ instead of the first preceding frame numbered as $f - 1$.

There are no intra updates allowed for B Frames. Four modes to predict blocks

of B-frames are: 1-intra prediction, 2-forward prediction, 3-backward prediction, 4-bidirectional prediction 5-direct prediction. The algorithm for intra and forward prediction is described in section 5.2. Direct prediction is a special case of bidirectional prediction where the forward motion vector is the half of the motion vector of the corresponding block in the P Frame. The backward motion vector is then just the opposite of the forward motion vector, i.e. for a direct predicted frame F with index f:

$$\begin{aligned} mv_{f_v}(f, b_v, b_h) &= \frac{mv_{f_v}(f+1, b''_v, b''_h)}{2}; \\ mv_{f_h}(f, b_h, b_v) &= \frac{mv_{f_h}(f+1, b''_v, b''_h)}{2}; \\ mv_{b_v}(f, b_v, b_h) &= -mv_{f_v}(f, b_v, b_h); \\ mv_{b_h}(f, b_h, b_v) &= -mv_{f_h}(f, b_v, b_h); \end{aligned}$$

We will investigate only the bidirectional prediction case for B frames. We had considered only the loss state of one preceding and the current frame for P-Frames. To model the decoder distortion for B frames we have to consider the loss states of the current, one preceding and one proceeding frame, i.e. we can consider eight different possibilities for the distortion of a block in the current frame F. In D_{xyz}, x gives information about the previous frame, y about the current and z about the next frame where $x, y, z \in l, r$. For example D_{lrr} denotes the distortion when the previous frame is lost, but the current one and the next ones are received. We will investigate the eight cases separately below:

Case 1

$D_{rrr(f,b_v,b_h)}$, the distortion of block b with coordinates (b_v, b_h) is given as follows:

$$\begin{aligned} D_{rrr}(f, b_v, b_h) &= \frac{\alpha(D_r(f-1, b'_v, b'_h) - D_q(f-1, b'_v, b'_h))}{4} \\ &+ \frac{\alpha^{\frac{3}{2}}(D_r(f-1, b'_v, b'_h) - D_q(f-1, b'_v, b'_h))}{2} \\ &+ \frac{\alpha(D_r(f+1, b''_v, b''_h) - D_q(f+1, b''_v, b''_h))}{4} \\ &+ D_q(f, b_v, b_h) \end{aligned} \tag{6.1}$$

The equation can easily be explained as follows: Let the error term on $b(f, b_v, b_h)$ be $d(f, b_v, b_h)$.

$$\begin{aligned} b(f, b_v, b_h) &= \frac{1}{2}\sqrt{\alpha}(b(f-1, b'_v, b'_h) + q(f-1, b'_v, b'_h) \\ &\quad + b(f+1, b''_v, b''_h) + q(f+1, b''_v, b''_h)) + mcbd(f, b_v, b_h) \end{aligned}$$

where $b(f-1, b'_v, b'_h)$ and $b(f+1, b''_v, b''_h)$ are the corresponding blocks in the forward and backward reference frames and $q(f-1, b'_v, b'_h)$ and $q(f+1, b''_v, b''_h)$ are their quantization errors respectively. According to this formula, b can be considered as an AR(1) source where $\sqrt{\alpha}$ corresponds to the correlation coefficient and $\sqrt{\alpha}(q(f-1, b'_v, b'_h) + q(f+1, b''_v, b''_h)) + mcbd(f, b_v, b_h)$ is the additional noise term. The noise term is assumed uncorrelated with $b(f, b_v, b_h)$, which is only the case if motion compensation is ideal. Denoting the accumulated error terms on $b(f, b_v, b_h)$, $b(f-1, b'_v, b'_h)$ and $b(f+1, b''_v, b''_h)$ as $d(f, b_v, b_h)$, $d(f-1, b'_v, b'_h)$ and $d(f+1, b''_v, b''_h)$ respectively, we have:

$$\begin{aligned} b(f, b_v, b_h) + d(f, b_v, b_h) &= \frac{1}{2}\sqrt{\alpha}(b'(f-1, b'_v, b'_h) + d'(f-1, b'_v, b'_h) \\ &\quad + b''(f+1, b''_v, b''_h) + d''(f+1, b''_v, b''_h)) \\ &\quad + mcbd(f, b_v, b_h) + q(f, b_v, b_h) \\ d(f, b_v, b_h) &= \frac{1}{2}\sqrt{(}\alpha)(d(f-1, b'_v, b'_h) - q(f-1, b'_v, b'_h) \\ &\quad + d(f+1, b''_v, b''_h) - q(f+1, b''_v, b''_h)) \\ &\quad + q(f, b_v, b_h) \end{aligned}$$

If the current and the previous frames are both received:

$$\begin{aligned} d(f-1, b'_v, b'_h) &= q(f-1, b'_v, b'_h) + d'_{rem} \\ d(f+1, b''_v, b''_h) &= q(f+1, b''_v, b''_h) + d''_{rem} \end{aligned}$$

where d'_{rem} and d''_{rem} are the components of $d(f-1, b'_v, b'_h)$ and $d(f+1, b''_v, b''_h)$ which are independent of $q(f-1, b'_v, b'_h)$ and $q(f+1, b''_v, b''_h)$. The correlation between

$d(f-1,b'_v,b'_h)$ and $q(f-1,b'_v,b'_h)$, similarly between $d(f+1,b''_v,b''_h)$ and $q(f+1,b''_v,b''_h)$ is given as:

$$\begin{aligned} E[d(f-1,b'_v,b'_h)q(f-1,b'_v,b'_h)] &= D_q(f-1,b'_v,b'_h) \\ E[d(f+1,b''_v,b''_h)q(f+1,b''_v,b''_h)] &= D_q(f+1,b''_v,b''_h) \end{aligned}$$

where $D_q(f-1,b'_v,b'_h)$ and $D_q(f+1,b''_v,b''_h)$ are the quantization distortions on the previous and next corresponding blocks respectively. Moreover $d(f+1,b''_v,b''_h)$ is given in terms of $d(f-1,b'_v,b'_h)$ as:

$$\begin{aligned} d(f+1,b''_v,b''_h) &= \sqrt{\alpha}(d(f-1,b'_v,b'_h) - q(f-1,b'_v,b'_h)) \\ &\quad +q(f+1,b''_v,b''_h) + mcbd(f+1,b''_v,b''_h); \\ d(f+1,b''_v,b''_h) - q(f+1,b''_v,b''_h) &= d(f-1,b'_v,b'_h) - q(f-1,b'_v,b'_h) + mcbd(f+1,b''_v,b''_h); \\ E[(d'_{rem}d''_{rem}] &= \sqrt{\alpha}(D_r(f-1,b'_v,b'_h) - D_q(f-1,b'_v,b'_h)); \end{aligned}$$

$D_{rrr}(f,b_v,b_h)$ is calculated as:

$$\begin{aligned} D_{rrr}(f,b_v,b_h) &= E[d(f,b_v,b_h)^2] \\ &= \frac{1}{4}\alpha(D_r(f-1,b'_v,b'_h) - D_q(f-1,b'_v,b'_h) \\ &\quad +D_r(f+1,b''_v,b''_h) - D_q(f+1,b''_v,b''_h) \\ &\quad +2\sqrt{\alpha}(D_r(f-1,b'_v,b'_h) - D_q(f-1,b'_v,b'_h))) \\ &\quad +D_q(f,b_v,b_h); \end{aligned}$$

yielding directly Formula 6.1.

Case 2

$D_{rrl}(f,b_v,b_h)$ is calculated in a similar way considering that the backward reference frame is lost. The only difference with case 1 is that:

$$E[(d(f+1,b''_v,b''_h)-q(f+1,b''_v,b''_h))^2] = D_l(f+1,b''_v,b''_h) + D_q(f+1,b''_v,b''_h)$$

since $d(f+1,b''_v,b''_h)$ and $q(f+1,b''_v,b''_h)$ are uncorrelated with each other.

Moreover we have:

$$d(f+1,b''_v,b''_h) = \sqrt{\alpha}(d(f-1,b'_v,b'_h)-q(f-1,b'_v,b'_h)) + mcbd(f+1,b''_v,b''_h)$$

yielding:

$$\begin{aligned} E[d(f+1,b''_v,b''_h)d'_{rem}] &= \sqrt{\alpha}(D_r(f-1,b'_v,b'_h)-D_q(f-1,b'_v,b'_h)) \\ E[q(f+1,b''_v,b''_h)d'_{rem}] &= 0 \end{aligned}$$

and resulting in the formula:

$$\begin{aligned} D_{rrl}(f,b_v,b_h) = & \frac{\alpha(D_r(f-1,b'_v,b'_h)-D_q(f-1,b'_v,b'_h))}{4} \\ & + \frac{\alpha^{\frac{3}{2}}(D_r(f-1,b'_v,b'_h)-D_q(f-1,b'_v,b'_h))}{2} \\ & + \frac{\alpha(D_r(f+1,b''_v,b''_h)+D_q(f+1,b''_v,b''_h))}{4} \\ & + D_q(f,b_v,b_h) \end{aligned}$$

Case3

Case 3 investigates the case that the previous frame is received and the current and the next frames are lost. The calculation is the same as for case 2 with the difference that the additional quantization distortion $D_q(f,b_v,b_h)$ is replaced by $pow\_mcbd(f,b_v,b_h)$, the averaged sum of the squared pixel intensities of the motion compensated block difference. The resulting decoder distortion formula is:

$$
\begin{aligned}
D_{rll}(f,b_v,b_h) &= \frac{\alpha(D_r(f-1,b'_v,b'_h)-D_q(f-1,b'_v,b'_h))}{4} \\
&+\frac{\alpha^{\frac{3}{2}}(D_r(f-1,b'_v,b'_h)-D_q(f-1,b'_v,b'_h))}{2} \\
&+\frac{\alpha(D_r(f+1,b''_v,b''_h)+D_q(f+1,b''_v,b''_h))}{4} \\
&+pow\_mcbd(f,b_v,b_h)
\end{aligned}
$$

Case 4

When the previous and next frames are received and the current one is lost, the formula we obtain is the same as in case 1 with the difference that the additional quantization distortion $D_q(f,b_v,b_h)$ is replaced by $pow\_mcbd(f,b_v,b_h)$, the averaged sum of the squared pixel intensities of the motion compensated block difference. The resulting decoder distortion formula is:

$$
\begin{aligned}
D_{rlr}(f,b_v,b_h) &= \frac{\alpha(D_r(f-1,b'_v,b'_h)-D_q(f-1,b'_v,b'_h))}{4} \\
&+\frac{\alpha^{\frac{3}{2}}(D_r(f-1,b'_v,b'_h)-D_q(f-1,b'_v,b'_h))}{2} \\
&+\frac{\alpha(D_r(f+1,b''_v,b''_h)-D_q(f+1,b''_v,b''_h))}{4} \\
&+pow\_mcbd(f,b_v,b_h)
\end{aligned}
$$

Case 5

When the previous and next frames are lost and the current one is received:

$$
\begin{aligned}
d(f,b_v,b_h) &= \frac{1}{2}\sqrt{\alpha}(d(f-1,b'_v,b'_h)-q(f-1,b'_v,b'_h) \\
&+d(f+1,b''_v,b''_h)-q(f+1,b''_v,b''_h)) \\
&+q(f,b_v,b_h) \\
d(f+1,b''_v,b''_h) &= \sqrt{\alpha}(d(f-1,b'_v,b'_h)-q(f-1,b'_v,b'_h))-mcbd(f+1,b''_v,b''_h)
\end{aligned}
$$

resulting in:

$$\begin{aligned} D_{lrl}(f,b_v,b_h) &= \frac{1}{4}\alpha(D_l(f-1,b'_v,b'_h)+D_q(f-1,b'_v,b'_h) \\ &\quad +D_l(f+1,b''_v,b''_h)+D_q(f+1,b''_v,b''_h) \\ &\quad +2E[(d(f-1,b'_v,b'_h)-q(f-1,b'_v,b'_h))d(f+1,b''_v,b''_h)]) \\ &\quad +D_q(f,b_v,b_h) \end{aligned}$$

since $d(f-1,b'_v,b'_h)$ is uncorrelated with $q(f-1,b'_v,b'_h)$ and similarly $d(f+1,b''_v,b''_h)$ is uncorrelated with $q(f+1,b''_v,b''_h)$. We see also:

$$E[d'_{rem}d(f+1,b''_v,b''_h)] = D_l(f-1,b'_v,b'_h)+D_q(f-1,b'_v,b'_h)$$

The resulting formula is then given as:

$$\begin{aligned} D_{lrl}(f,b_v,b_h) &= (\frac{1}{4}\alpha+\frac{1}{2}\alpha^{\frac{3}{2}})(D_l(f-1,b'_v,b'_h)+D_q(f-1,b'_v,b'_h)) \\ &\quad +\frac{1}{4}\alpha(D_l(f+1,b''_v,b''_h)+D_q(f+1,b''_v,b''_h)) \\ &\quad +D_q(f,b_v,b_h) \end{aligned}$$

Case 6

Next, we explore the case that the previous and the current frames are lost and the next one is received. This time $d(f-1,b'_v,b'_h)$ is uncorrelated with $q(f-1,b'_v,b'_h)$ but $d(f+1,b''_v,b''_h)$ is correlated with $q(f+1,b''_v,b''_h)$ giving:

$$\begin{aligned} D_{llr}(f,b_v,b_h) &= \frac{1}{4}\alpha(D_l(f-1,b'_v,b'_h)-D_q(f-1,b'_v,b'_h) \\ &\quad +D_r(f+1,b''_v,b''_h)+D_q(f+1,b''_v,b''_h) \\ &\quad +2E[d'_{rem}d''_{rem}] \\ &\quad +pow\_mcbd(f,b_v,b_h) \end{aligned}$$

moreover

$$
\begin{aligned}
d(f+1, b''_v, b''_h) &= \sqrt{\alpha}(d(f-1, b'_v, b'_h) - q(f-1, b'_v, b'_h)) + q(f+1, b''_v, b''_h) \\
E[d'_{rem} d''_{rem}] &= \sqrt{\alpha}(D_l(f-1, b'_v, b'_h) - D_q(f-1, b'_v, b'_h)
\end{aligned}
$$

giving the formula:

$$
\begin{aligned}
D_{llr}(f, b_v, b_h) &= (\frac{1}{4}\alpha + \frac{1}{2}\alpha^{\frac{3}{2}})(D_l(f-1, b'_v, b'_h) + D_q(f-1, b'_v, b'_h)) \\
&+\frac{1}{4}\alpha(D_r(f+1, b''_v, b''_h) - D_q(f+1, b''_v, b''_h)) \\
&+pow\_mcbd(f, b_v, b_h)
\end{aligned}
$$

Case 7

When the previous frame is lost and the current and the next frames are received:

$$
\begin{aligned}
D_{lrr}(f, b_v, b_h) &= (\frac{1}{4}\alpha + \frac{1}{2}\alpha^{\frac{3}{2}})(D_l(f-1, b'_v, b'_h) + D_q(f-1, b'_v, b'_h)) \\
&+\frac{1}{4}\alpha(D_r(f+1, b''_v, b''_h) - D_q(f+1, b''_v, b''_h)) \\
&+D_q(f, b_v, b_h)
\end{aligned}
$$

Case 8

When the previous, current and the next frames are lost:

$$
\begin{aligned}
D_{lll}(f, b_v, b_h) &= (\frac{1}{4}\alpha + \frac{1}{2}\alpha^{\frac{3}{2}})(D_l(f-1, b'_v, b'_h) + D_q(f-1, b'_v, b'_h)) \\
&+\frac{1}{4}\alpha(D_l(f+1, b''_v, b''_h) + D_q(f+1, b''_v, b''_h)) \\
&+pow\_mcbd(f, b_v, b_h)
\end{aligned}
$$

with the similar argumentation and calculations as above.

$D_l(f, b_v, b_h)$ and $D_r(f, b_v, b_h)$ are then given as the weighted averages of the estimated distortion terms denoting the distortion terms that the current frame was lost and received respectively:

	QP first	QP remain.	QP B
F.	14	14	14
A.	18	17	18
C.	19	19	18
M&D.	15	15	14
S.	15	15	14

Table 6.1: TLC, Quantization Parameters

$$\begin{aligned} D_l(f,b_v,b_h) &= p_l^2 D_{lll}(f,b_v,b_h) + (1-p_l)p_l D_{rll}(f,b_v,b_h) \\ &\quad +(1-p_l)^2 D_{rlr}(f,b_v,b_h) + (1-p_l)p_l D_{llr}(f,b_v,b_h) \\ D_r(f,b_v,b_h) &= p_l^2 D_{lrl}(f,b_v,b_h) + (1-p_l)p_l D_{rrl}(f,b_v,b_h) \\ &\quad +(1-p_l)^2 D_{rrr}(f,b_v,b_h) + (1-p_l)p_l D_{lrr}(f,b_v,b_h) \end{aligned}$$

PSNR values are calculated directly from the distortion terms as:

$$PSNR_{xyz}(f) = 10log10\frac{255^2}{\frac{\sum_{i=1}^{max_i}\sum_{j=1}^{max_j} D_{xyz}(f,i,j)}{total\_block\_number}}$$

where xyz is the index of the corresponding distortion term. The overall PSNR is the average of all PSNR terms weighted by their respective probabilities.

6.1.1 without intra-updates

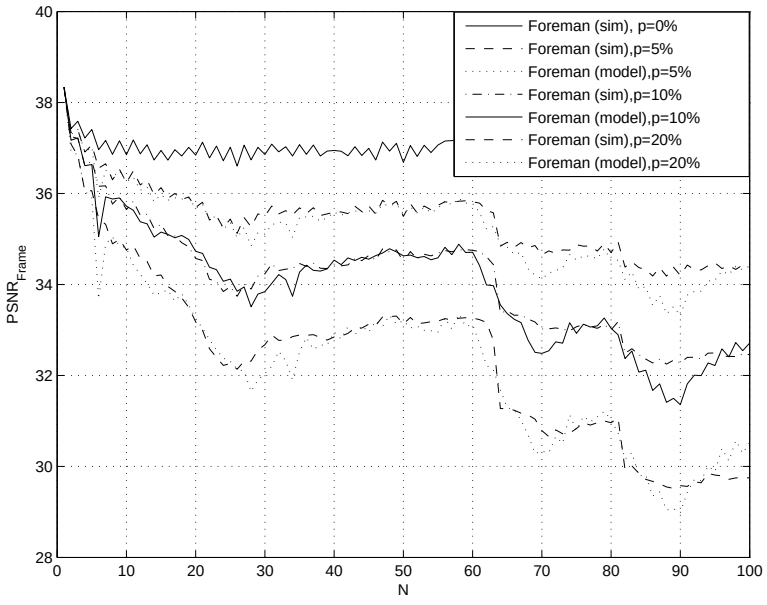
Table 6.1 gives the quantization step sizes for I, P and B frames for sequences Foreman, Akiyo, Claire, Mother & Daughter and Salesman. Table 6.2, on the other hand, lists the coding rate and coded frame PSNR when no-intra coding is used.

Figures 6.1, 6.2, 6.3, 6.4 and 6.5 show the model-simulation correspondence for Foreman, Akiyo, Claire, Mother & Daughter and Salesman respectively. The zigzag form of the curve is due to the quantization differences between P and B frames. The same α value is used both for P and B frames.

Figure 6.6 shows the course of α for the sequences. Course of α for no-intra-updates differentiates a little between single layered and temporal layered coding as

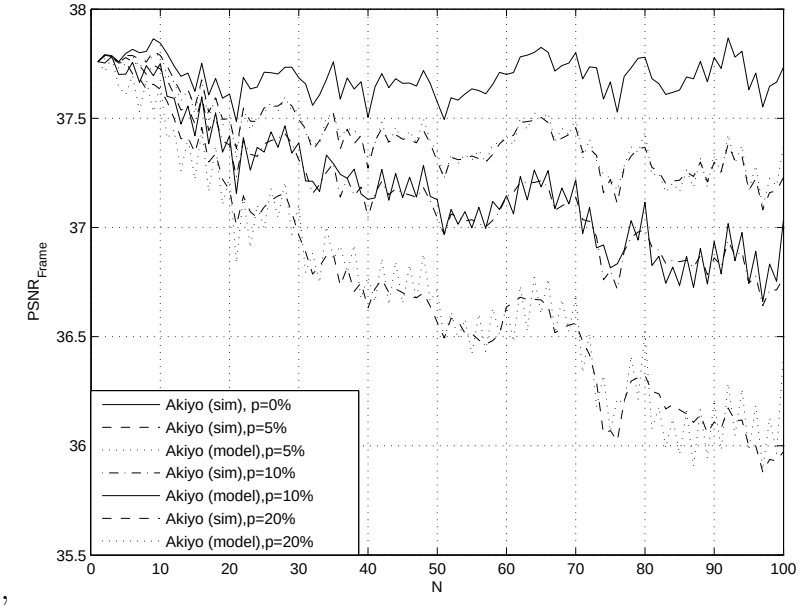
	avg. PSNR [dB] no-i.-updates	Bitrate [kbits/s] no-i.-updates	i.-GOB period	i.-frame period
F.	37.02	160.86	0	0
A.	37.67	18.15	0	0
C.	37.98	21.79	0	0
M&D.	37.40	73.92	0	0
S.	36.51	65.45	0	0

Table 6.2: TLC, no-intra-coding Parameters



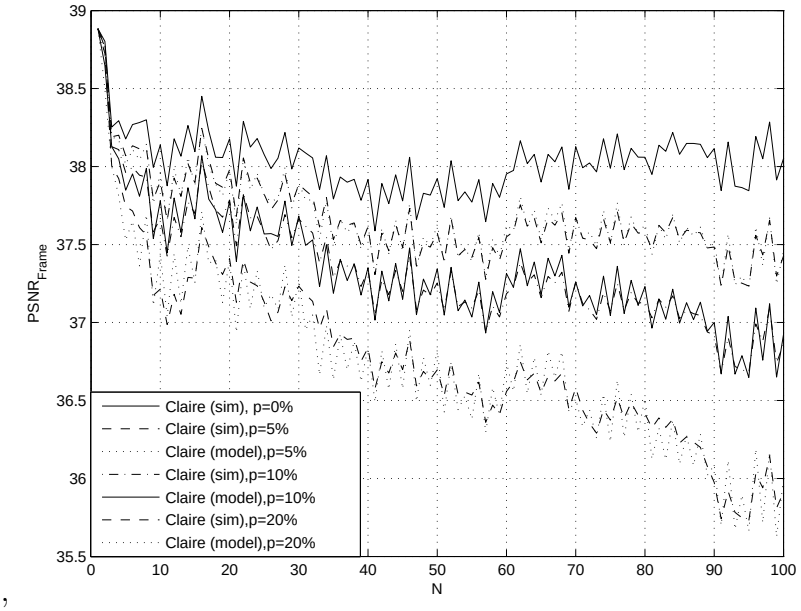
,

Figure 6.1: TLC, PSNR over Frame Number, Model-Simulation Accordance, Foreman, no-intra-updates



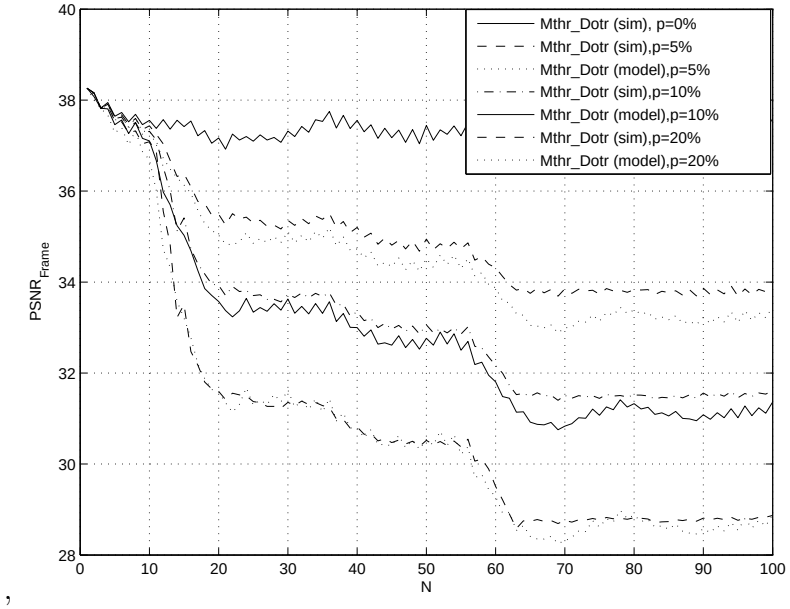
,

Figure 6.2: TLC, PSNR over Frame Number, Model-Simulation Accordance, Akiyo, no-intra-updates



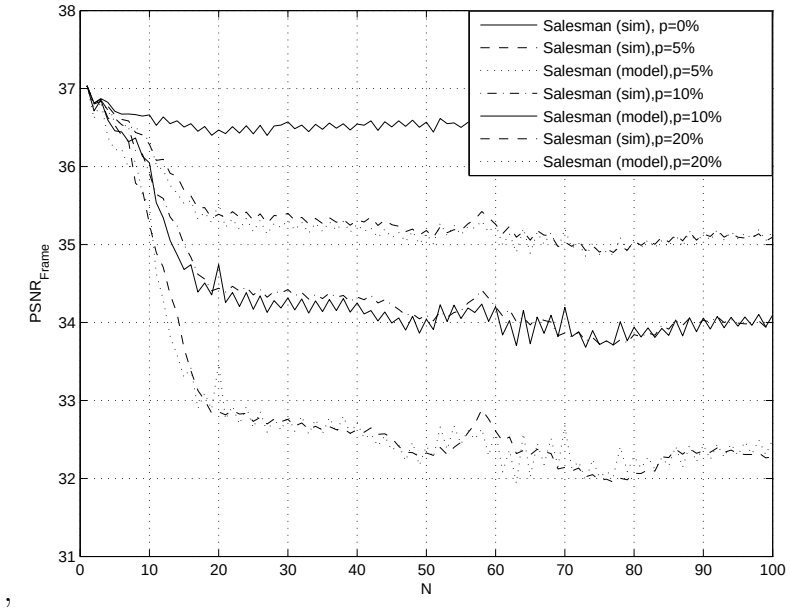
,

Figure 6.3: TLC, PSNR over Frame Number, Model-Simulation Accordance, Claire, no-intra-updates



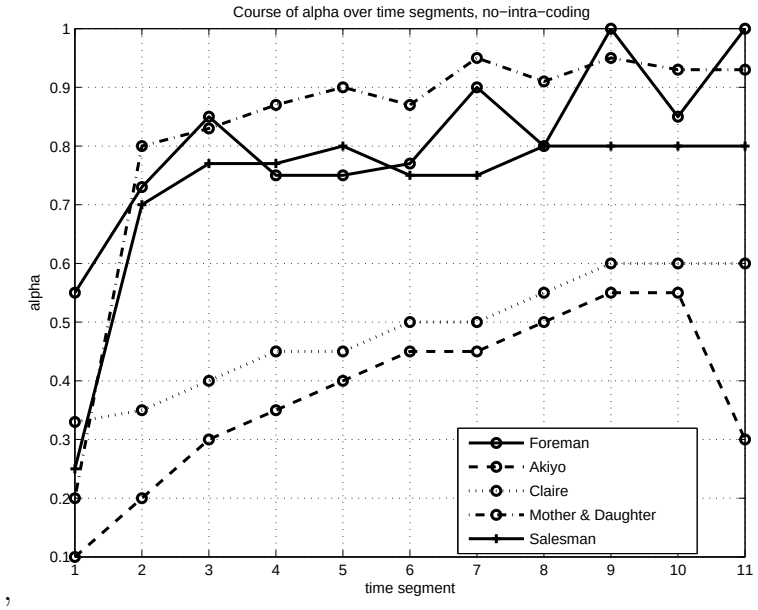
,

Figure 6.4: TLC, PSNR over Frame Number, Model-Simulation Accordance, Mother & Daughter, no-intra-updates



,

Figure 6.5: TLC, PSNR over Frame Number, Model-Simulation Accordance, Salesman, no-intra-updates



,

Figure 6.6: TLC, Course of α over time segments, no-intra-updates

	avg. PSNR [dB] i.-GOB-updates	**Bitrate [kbits/s] i.-GOB-updates**	**i.-GOB period**	**i.-frame period**
F.	36.99	197.55	1	0
A.	37.38	47.89	1	0
C.	38.12	38.87	1	0
M&D.	37.31	105.63	1	0
S.	36.01	121.44	1	0

Table 6.3: TLC, intra-GOB-coding Parameters

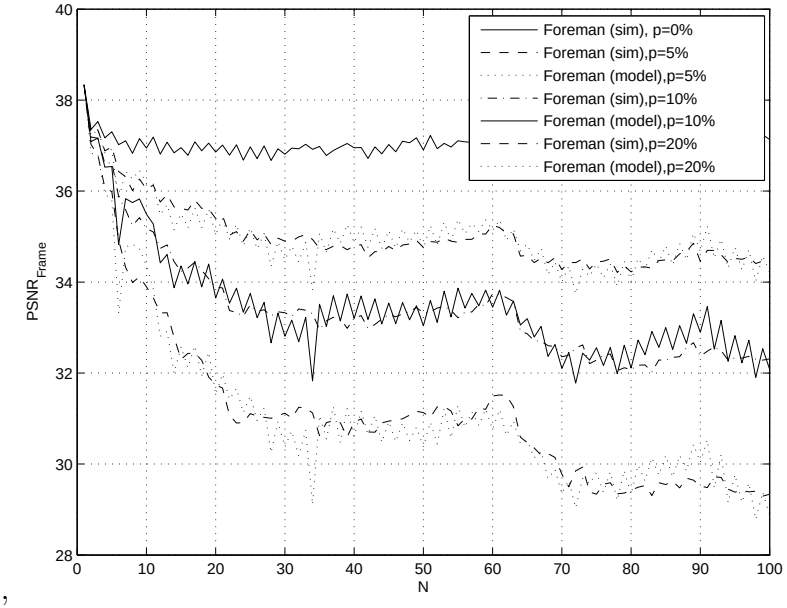
seen from a comparison of Figures 5.8 and 6.6.

6.1.2 with intra-GOB-updates

Table 6.3 gives the coding rate and encoded frame PSNR when in each frame periodically one GOB is updated.

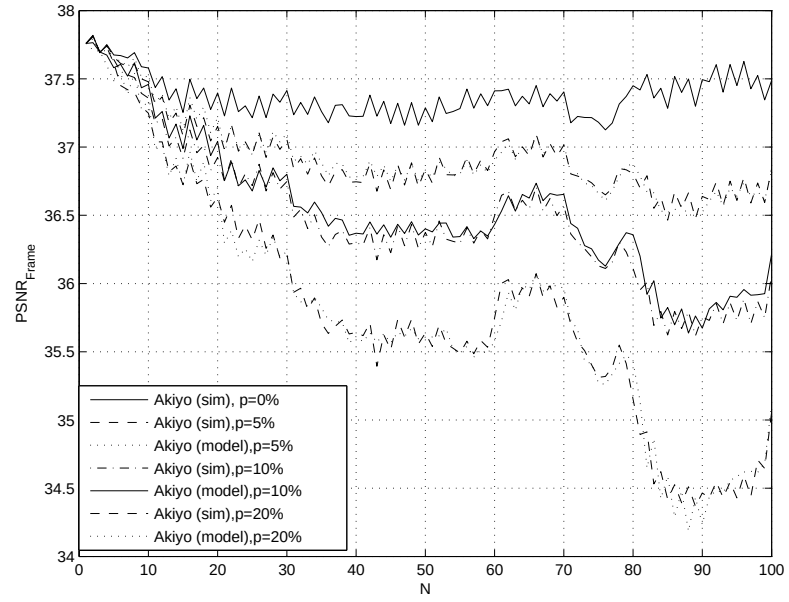
Figures 6.7, 6.8, 6.9, 6.10 and 6.11 show the temporal layered decoder distortion estimation with intra-GOB-updates.

The course of α over time segments with intra-GOB's is shown in Figure 6.12. When compared to the course of α for single layered coding with intra-GOB-updates seen in Figure 6.18, the form and relation of curves to each other are very similar.



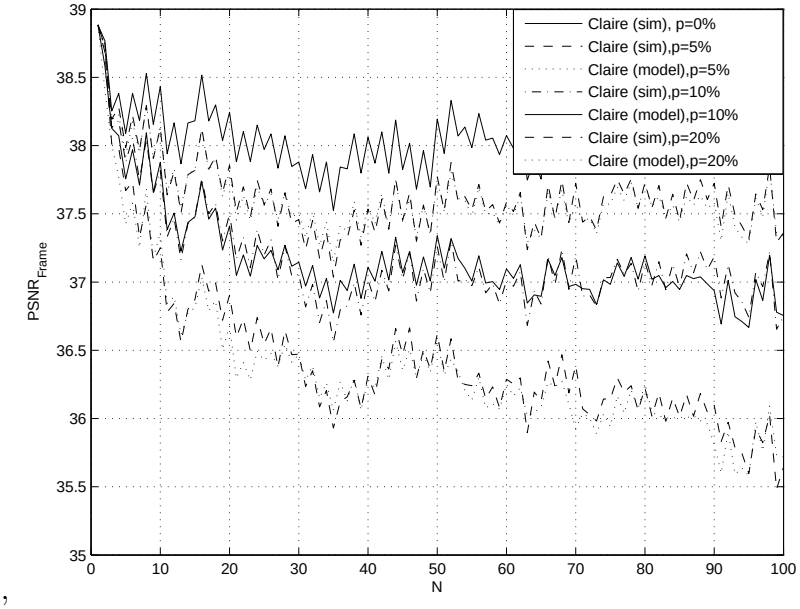
,

Figure 6.7: TLC, PSNR over Frame Number, Model-Simulation Accordance, Foreman, intra-GOB-updates



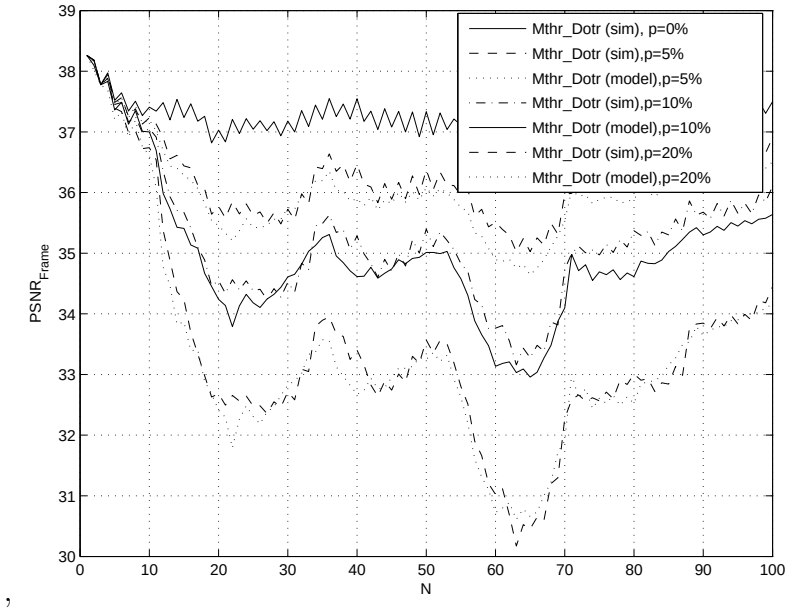
,

Figure 6.8: TLC, PSNR over Frame Number, Model-Simulation Accordance, Akiyo, intra-GOB-updates



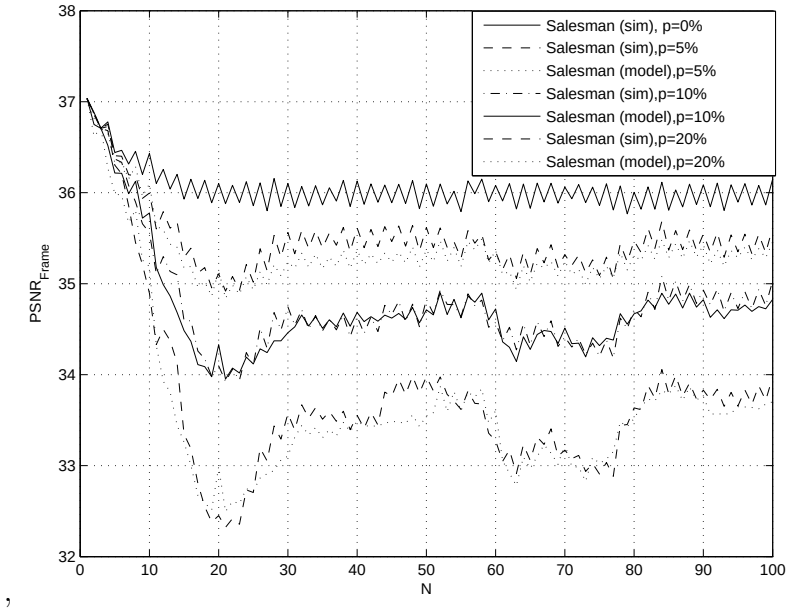
,

Figure 6.9: TLC, PSNR over Frame Number, Model-Simulation Accordance, Claire, intra-GOB-updates



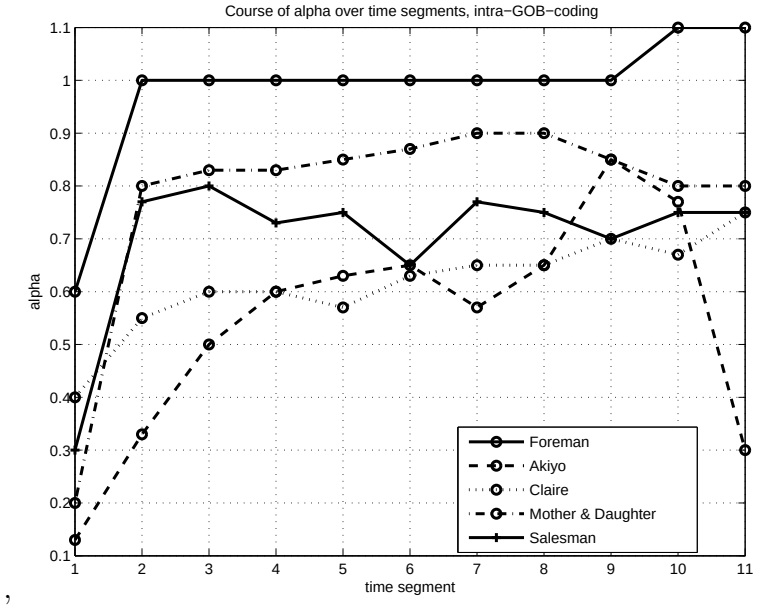
,

Figure 6.10: TLC, PSNR over Frame Number, Model-Simulation Accordance, Mother & Daughter, intra-GOB-updates



,

Figure 6.11: TLC, PSNR over Frame Number, Model-Simulation Accordance, Salesman, intra-GOB-updates

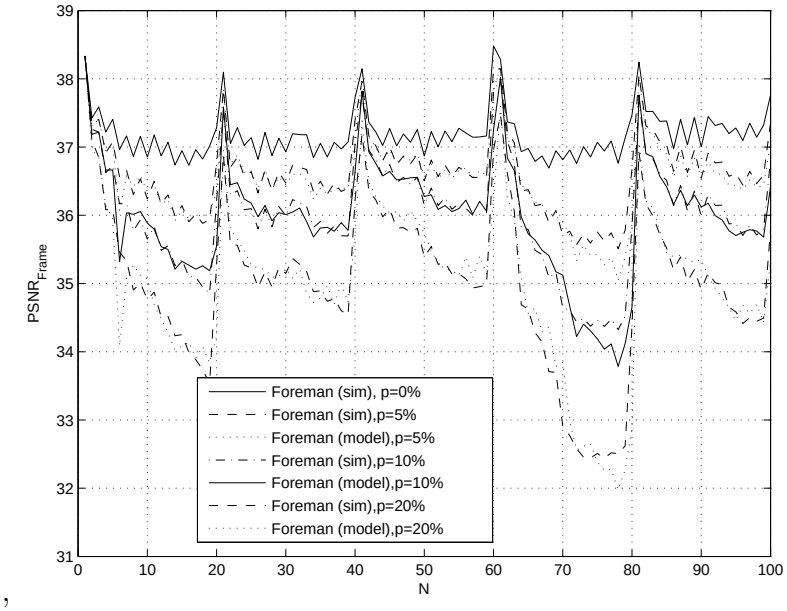


,

Figure 6.12: TLC, Course of α over time segments, intra-GOB-coding

	avg. PSNR [dB] i.-frame-updates	Bitrate [kbits/s] i.-frame-updates	i.-GOB period	i.-frame period
F.	37.22	194.54	0	10
A.	37.79	42.54	0	10
C.	38.50	34.37	0	10
M&D.	37.73	99.00	0	10
S.	36.72	106.17	0	10

Table 6.4: TLC, intra-frame-coding Parameters



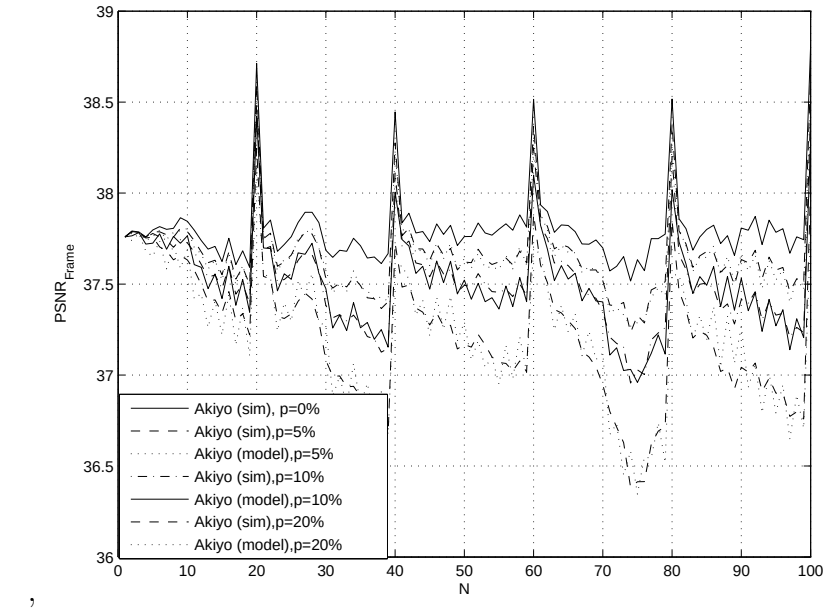
,

Figure 6.13: TLC, PSNR over Frame Number, Model-Simulation Accordance, Foreman, intra-frame-updates

6.1.3 with intra-frame-updates

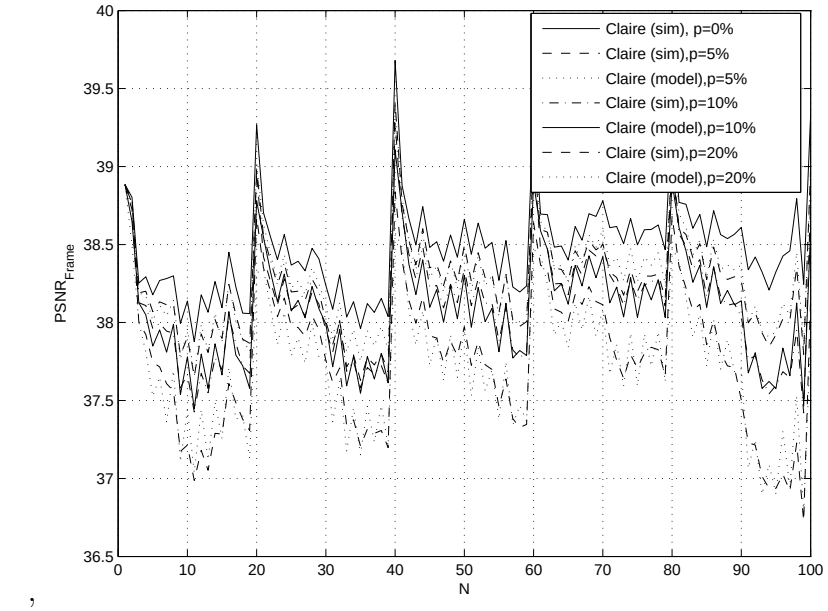
The bitrates and encoded frame PSNR's for intra-frame-coded temporal layered coding are listed in Table 6.4. Intra-frame-updates occur at every tenth frame. Figures 6.13, 6.14, 6.15, 6.16 and 6.17 show the PSNR over frame number for all the sequences respectively and confirm the good correspondence of model and simulation results.

Change of α over time is depicted in Figure 6.18 for intra-frame-updated and temporal layered coded sequences. Again a comparison with Figure 5.20 shows the similarity between the α values of temporal layered and single layered coding using the same coding option, i.e. here intra-frame-updates with the same period.



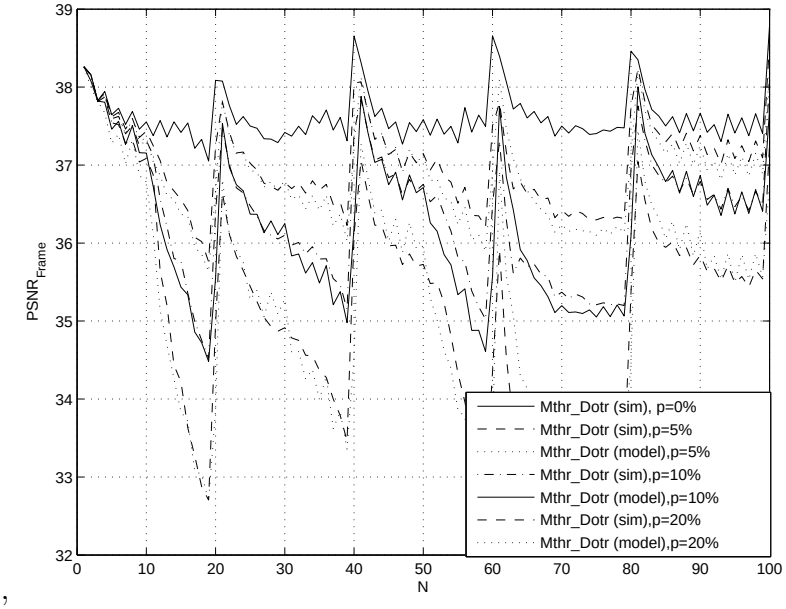
,

Figure 6.14: TLC, PSNR over Frame Number, Model-Simulation Accordance, Akiyo, intra-frame-updates



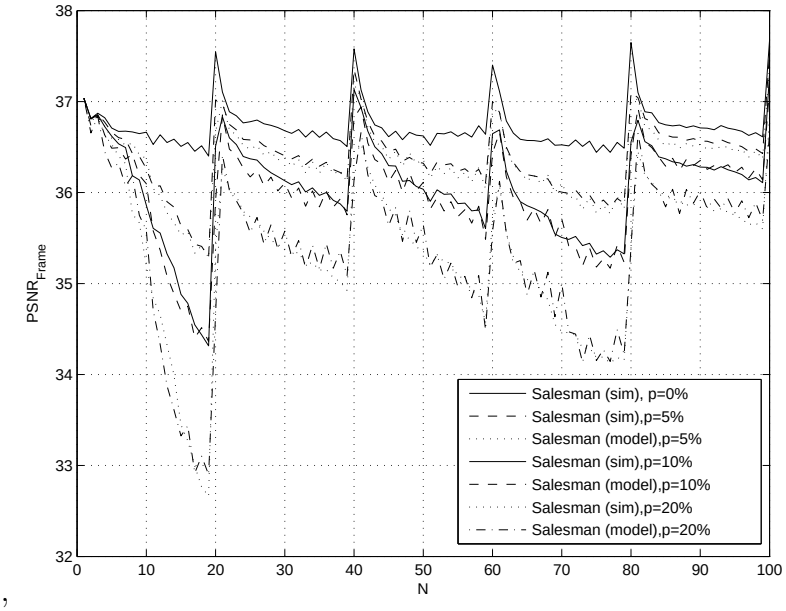
,

Figure 6.15: TLC, PSNR over Frame Number, Model-Simulation Accordance, Claire, intra-frame-updates



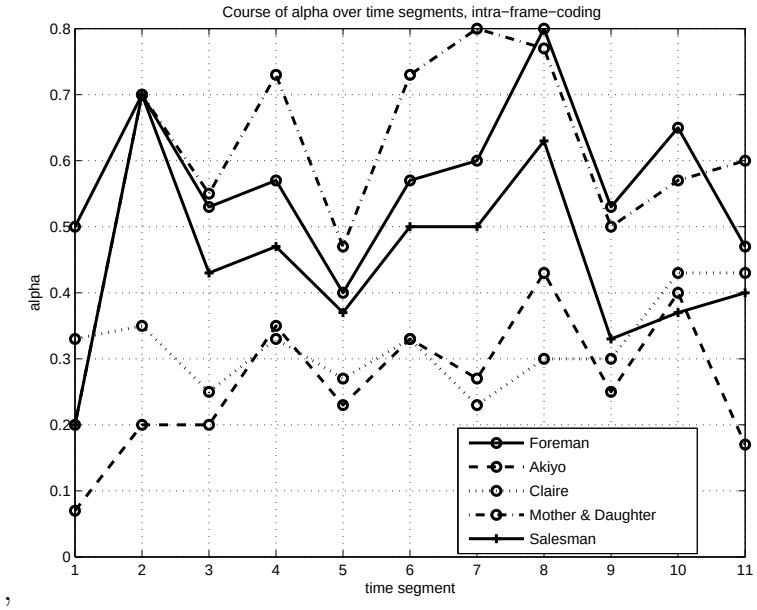
,

Figure 6.16: TLC, PSNR over Frame Number, Model-Simulation Accordance, Mother & Daughter, intra-frame-updates



,

Figure 6.17: TLC, PSNR over Frame Number, Model-Simulation Accordance, Salesman, intra-frame-updates



,

Figure 6.18: TLC, Course of α over time segments, intra-frame-updates

Next, for each sequence Foreman, Akiyo, Claire, Mother & Daughter and Salesman, the course of α for the three coding options: no-intra-coding, intra-GOB-coding and intra-frame-coding are compared to each other in Figures 6.19, 6.20, 6.21, 6.22 and 6.23 respectively. As in case of single layered coding, the relation of α curves corresponding to different coding options remains the same. For Foreman, Akiyo and Claire, the GOB-intra-updates require the highest α, for Mother & Daughter and Salesman, on the other hand, the no-intra-coding requires slightly bigger values. The frame-intra-updated sequences need the lowest α values for all sequences since there is no error accumulation due to previous time segments. Since the frame update period is as long as the time segment, the form of the α curve is directly related to the changes of the spatial and temporal activities in the sequence.

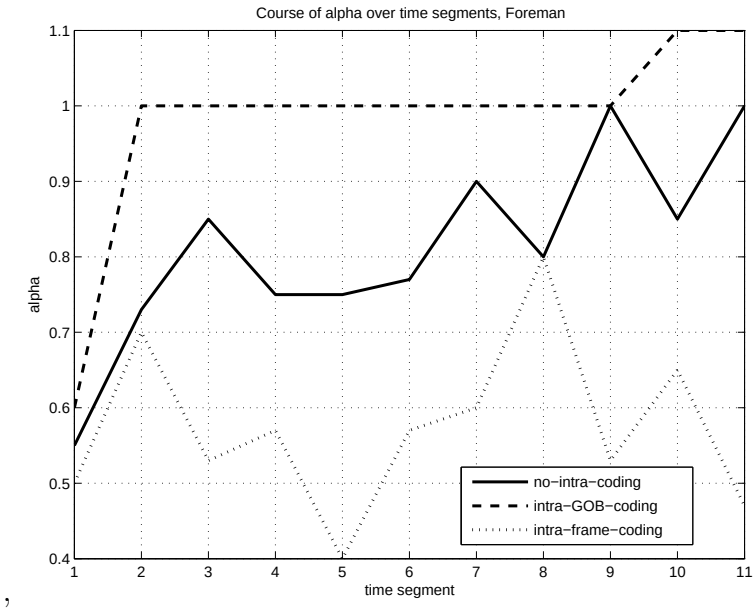


Figure 6.19: TLC, Course of α over time segments, Foreman

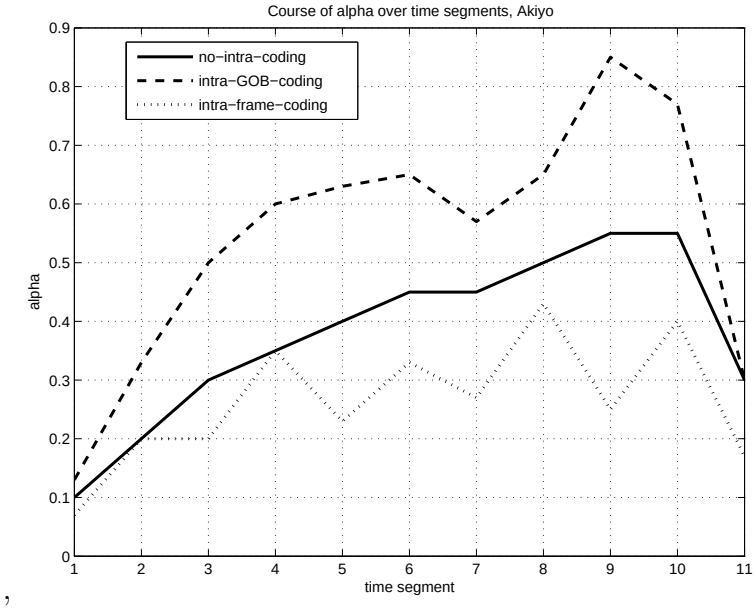
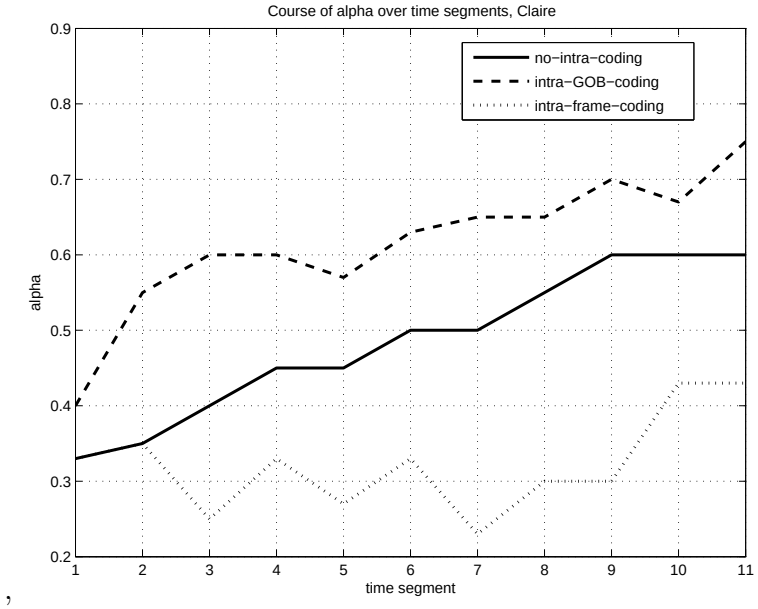
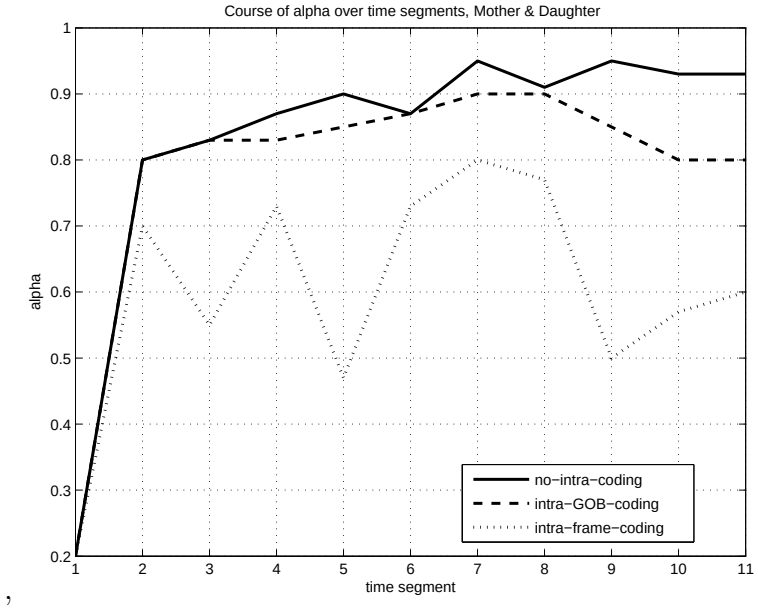


Figure 6.20: TLC, Course of α over time segments, Akiyo



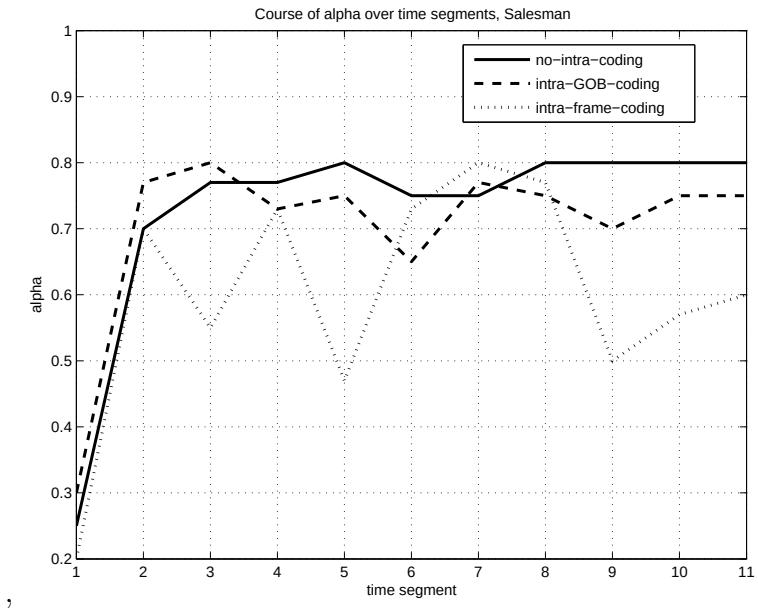
,

Figure 6.21: TLC, Course of α over time segments, Claire



,

Figure 6.22: TLC, Course of α over time segments, Mother & Daughter



,

Figure 6.23: TLC, Course of α over time segments, Salesman

Chapter 7

Decoder Distortion Estimation for Multi-State Video Coding

The decoder distortion estimation method discussed for single-layered and temporal-layered transmission in chapters 5 and 6 is extended here for multi-state video coding. Section 7.1 investigates recursive distortion estimation for a one-dimensional AR(1) source and multi-state coding and transmission. Section 7.2 extends the ideas presented in 7.1 to video signals where the corresponding blocks in subsequent frames along the sequence are modeled as samples from the same AR(1) source. The work is presented in [36].

7.1 Recursive Distortion Estimation for a one-dimensional AR(1) source

In this section we analyze the distortion occurring during the multi-stream transmission of a one-dimensional AR(1) source. We assume having two transmission paths with independent loss probabilities p_1 and p_2. The signal to be transmitted $x(n)$ is split into two, where the odd samples are transmitted over the channel with loss probability p_1 whereas the even samples are sent over the other path and lost with probability p_2. The lost samples are substituted by bilinear interpolation of their previous and next samples. The two streams constituted by the odd and even samples respectively are coded with different bitrates R_1 and R_2 resulting in quantization error variances σ_{q1}^2 and σ_{q2}^2. Our aim is to model the average distortion accumulated

on each stream due to losses and associated concealments through frame interpolation/repetition, as well as due to quantization errors. Modeled average distortions over all possible bitrate allocations (R_1, R_2) (where $R_T = R_1 + R_2$ for different combinations of (p_1, p_2)) will provide insight into the optimal bitrate allocation under different transmission conditions.

The AR(1) source is described by $x(n) = a_1 x(n-1) + z(n)$ where $a_1 = \rho_1$ is the normalized autocorrelation coefficient of the source and $z(n)$ is a zero mean uncorrelated source. According to that, the odd and even splitted sources can be described by:

$$\begin{aligned} x_1(n_1) &= a_2 x_1(n_1) + z_1(n_1) \\ x_2(n_1) &= a_2 x_2(n_1) + z_2(n_1) \end{aligned}$$

where $a_2 = a_1^2$ and $z_1(n_1)$ and $z_2(n_1)$ are zero mean uncorrelated sources. To simplify further calculations without loss of generality we will assume that $\sigma_x^2 = 1$. $x(n)$ and $z(n)$ have zero means and $\sigma_z^2 = 1 - a_1^2$. $z_1(n_1)$ and $z_2(n_1)$ are odd and even splitted parts of $z(n)$ respectively. Differential Pulse Code Modulation (DPCM) with optimal prediction is used to code the samples, i.e.:

$$\begin{aligned} d_1(n_1) &= x_1(n_1) - a_2 x_1(n_1 - 1) \\ d_2(n_1) &= x_2(n_1) - a_2 x_2(n_1 - 1) \end{aligned}$$

Our goal is to estimate the accumulated distortion recursively on each sample given (R_1, R_2) and (p_1, p_2). We verify the model via simulations generating $x(n)$ as an AR(1) source with the predefined a_1 and simulating the transmissions with loss probabilities p_1 and p_2 in section 7.3.
Below we discuss different cases due to the reception or lost of current and neighbor samples.

Estimation of Distortion if sample x(n) is lost:

Suppose the sample $x(n)$ on the second stream is lost and has to be interpolated using $x(n-1)$ and $x(n+1)$. The error on the n^{th} sample $e(n)$ needs to be estimated. $e(n)$ depends on $e(n-1)$ and $e(n+1)$ as well as on σ_{interp}^2 which is a function of a_1. $D(n) = \sigma_{interp}^2$ when $e(n-1) = e(n+1) = 0$. Assuming bilinear interpolation $x(n) =$

$\frac{x(n-1)+x(n+1)}{2}$, the error $e(n)$ is given as $e(n) = \frac{e(n-1)+e(n+1)}{2} + e_{interp}$. We assume that e_{interp} is independent of the rest of the error term. The samples $x(n+1)$ and $x(n-1)$ are received with probability of $(1-p_1)$. The error terms $e(n-1)$ and $e(n+1)$ include not only the quantization errors but also the through error propagation accumulated errors. Besides, $e(n+1)$ can be formulated in terms of $e(n-1)$:

$$e(n+1) = a_2(e(n-1) - q_1(n-1)) + q_1(n+1)$$

where $q_1(n-1)$ and $q_1(n+1)$ are the quantization errors on $x(n-1)$ and $x(n+1)$ respectively (subscript 1 denotes the first stream). The error $q_1(i-1)$ is compensated on the prediction loop so that the sender and the receiver work with the same prediction values if no losses occur. Moreover the through error propagation accumulated residual error is filtered. $e(n)$ is then calculated as:

$$\begin{aligned} e(n) &= \frac{e(n-1)+e(n+1)}{2} + e_{interp} \\ &= \frac{(1+a_2)e(n-1) - a_2 q_1(n-1) + q_1(n+1)}{2} + e_{interp} \end{aligned} \tag{7.1}$$

Next we calculate the distortion $D(n)$ on $x(n)$:

$$\begin{aligned} D(n) &= E[e^2(n)] \\ &= \frac{(1+a_2)^2 D(n-1) + a_2^2\sigma_{q1}^2 + \sigma_{q1}^2 - 2a_2(1+a_2)(1-p_1)\sigma_{q1}^2}{4} + \sigma_{interp}^2 \\ &= \frac{(1+a_2)^2}{4} D(n-1) + \frac{1+a_2^2}{4}\sigma_{q1}^2 - \frac{a_2(1+a_2)(1-p_1)\sigma_{q1}^2}{2} + \sigma_{interp}^2 \end{aligned} \tag{7.2}$$

If $x(n-1)$ is received with a probability of $1-p_1$, $e(n-1)$ has a component equal to $q_1(n-1)$ whereas the residual error $e(n-1) - q_1(n-1)$ is uncorrelated with $q_1(n-1)$, yielding:

$$E[e(n-1)q(n-1)] = (1-p_1)\sigma_{q1}^2$$

Moreover σ^2_{interp} can be calculated as:

$$\begin{aligned}
\hat{x}(n) &= \frac{x(n-1)+x(n+1)}{2} \\
&= \frac{x(n-1)+a_1x(n)+z(n+1)}{2} \\
e_{interp} &= x(n)-\hat{x}(n) \\
&= x(n)-\frac{a_1}{2}x(n)-\frac{z(n+1)}{2}-\frac{x(n-1)}{2} \\
e_{interp} &= x(n)(1-\frac{a_1}{2})-\frac{x(n-1)}{2}-\frac{z(n+1)}{2} \\
\sigma^2_{interp} &= E[e^2_{interp}] \\
&= (1-\frac{a_1}{2})^2+\frac{1}{4}+\frac{1-a_1^2}{4}-a_1(1-\frac{a_1}{2}) \\
&= (1-a_1)^2+\frac{1}{2}(1-a_1^2) \qquad (7.3)
\end{aligned}$$

where $\hat{x}(n)$ is the interpolated sample.

Estimation of Distortion if samples x(n) and x(n-2) are both received:

x(n) is reconstructed as $\tilde{x}(n)$:

$$\begin{aligned}
\tilde{x}(n) &= a_2\tilde{x}(n-2)+d(n)+q_2(n) \\
e(n) &= a_2(e(n-2)-q_2(n-2))+q_2(n) \\
D(n) &= E[e^2(n)] = a_2^2D(n-2)+a_2^2\sigma^2_{q2}+\sigma^2_{q2}-2a_2^2\sigma^2_{q2} \\
&= a_2^2D(n-2)+(1-a_2^2)\sigma^2_{q2} \qquad (7.4)
\end{aligned}$$

where $d(n)$ is the difference value sent to the receiver and σ^2_{q2} is the quantization error variance of the second stream where $x(n)$ was coded and transmitted.

Estimation of Distortion if sample x(n) is received but x(n-2) is lost:

$$\begin{aligned}
\tilde{x}(n) &= a_2\tilde{x}(n-2)+d(n)+q_2(n) \\
e(n) &= a_2(e(n-2)-q_2(n-2))+q_2(n) \\
D(n) &= E[e^2(n)] = a_2^2D(n-2)+a_2^2\sigma^2_{q2}+\sigma^2_{q2}-2a_2^2E[e(n-2)q_2(n-2)] \\
&= a_2^2D(n-2)+(1+a_2^2)\sigma^2_{q2} \qquad (7.5)
\end{aligned}$$

Estimation of Distortion if sample x(n) is lost but x(n+1) is received:

$\tilde{x}(n)$ is constructed by interpolating $\tilde{x}(n-1)$ and $\tilde{x}(n+1)$.

$$\begin{aligned}
\tilde{x}(n) &= \frac{\tilde{x}(n-1)+\tilde{x}(n+1)}{2} \\
&= \frac{x(n-1)+x(n+1)+e(n-1)+e(n+1)}{2} \\
e(n) &= x(n) - \frac{x(n-1)+x(n+1)+e(n-1)+e(n+1)}{2} \\
&= e_{interp} + \frac{e(n-1)+e(n+1)}{2} \\
&= e_{interp} + \frac{e(n-1)+a_1^2 e(n-1) - a_1^2 q_1(n-1) + q_1(n+1)}{2} \\
D(n) &= \sigma^2_{interp} + \frac{(1+a_1^2)^2}{4} D(n-1) + \frac{(1+a_1^4)}{4}\sigma^2_{q1} - (1-p_1)\frac{(1+a_1^2)a_1^2}{2}\sigma^2_{q1}
\end{aligned} \tag{7.6}$$

where $E[e(n-1)q(n-1)] = (1-p_1)\sigma^2_{q1}$ and $(1-p_1)$ is the probability that $x(n-1)$ was not lost.

Estimation of Distortion if samples x(n) and x(n+1) are both lost:

The reconstruction method in this case is the repetition of the last sample.

$$\begin{aligned}
\tilde{x}(n) &= \tilde{x}(n-1) \\
e(n) &= e(n-1) + e_{rep} \\
D(n) &= D(n-1) + \sigma^2_{rep}
\end{aligned} \tag{7.7}$$

with

$$\begin{aligned}
e_{rep}(n) &= x(n) - x(n-1) \\
\sigma^2_{rep} &= E[e^2_{rep}(n)] = 1 + 1 - 2a_1 = 2(1-a_1)
\end{aligned} \tag{7.8}$$

Here we assumed that $e(n-1)$ is not correlated to $x(n-1)$ or $x(n)$ which does not hold as p_1, p_2 and/or n is increasing. The model can be improved by taking the correlation with the error term into account.

The overall distortion model for the n^{th} sample (on the second stream) can be summarized as:

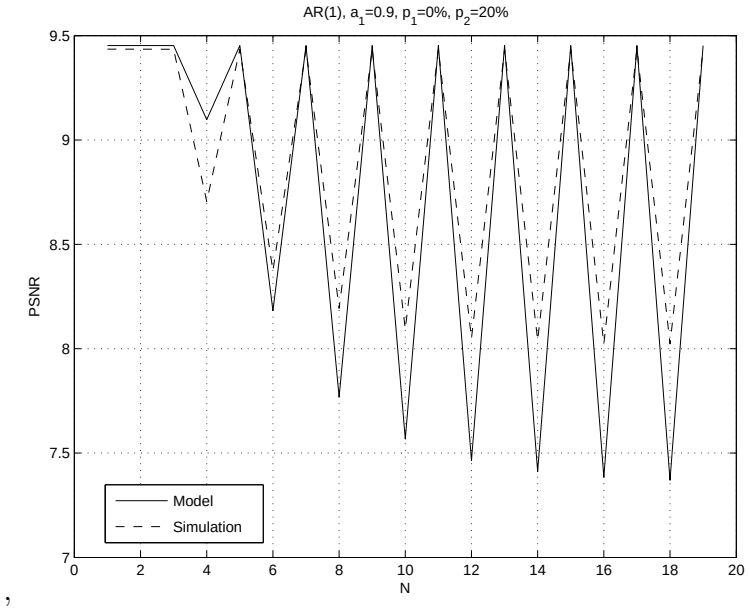
$$
\begin{aligned}
D(n) &= (1-p_2)(1-p_2)(a_2^2 D(n-2) + (1-a_2^2)\sigma_{q2}^2) \\
&\quad +(1-p_2)p_2(\frac{3}{2}a_1^4 + \frac{1}{4}a_1^4 D(n-3) + \frac{1}{4}a_1^4 D(n-1) \\
&\quad +(1+a_1^4)\sigma_{q2}^2 - 2a_1^5 + \frac{1}{2}a_1^6[1+(1-p_1)D(n-3)]) \\
&\quad +p_2(1-p_1)(\sigma_{interp}^2 + \frac{(1+a_1^2)^2}{4}D(n-1) + \frac{(1+a_1^4)}{4}\sigma_{q1}^2 \\
&\quad -(1-p_1)\frac{(1+a_1^2)a_1^2}{2}\sigma_{q1}^2) + p_2 p_1(D(n-1) + \sigma_{rep}^2) \qquad (7.9)
\end{aligned}
$$

For a sample on the first stream we obtain the symmetrical equation when p_1 is exchanged with p_2 and σ_{q1}^2 with σ_{q2}^2. The first part of Equation 7.9 considers the case that the n^{th} and the $(n-2)^{th}$ samples both are received. The second part belongs to the case that the n^{th} sample is received and the $(n-2)^{th}$ sample is lost whereas, the third part to the case that the n^{th} lost and the $(n+1)^{th}$ is received. The last and fourth line deal with the case that both n^{th} and $(n+1)^{th}$ are lost.

Figures 7.1 and 7.2 compare the results obtained via model and simulations. For simulations one-dimensional AR(1) sources with correlation coefficient $a_1 = 0.9$ and variance $\sigma_x^2 = 1$ are generated and multi-stream transmission with $R = 0.8 bit/sample$ is simulated using 1000 different random loss patterns for each loss probability p_1 and p_2. As the loss probabilities p_1 and p_2 increase the difference between the simulation and model results increases for large n. The main reason for this gap is the uncorrelation assumption of different error terms which is not valid for increasing N.

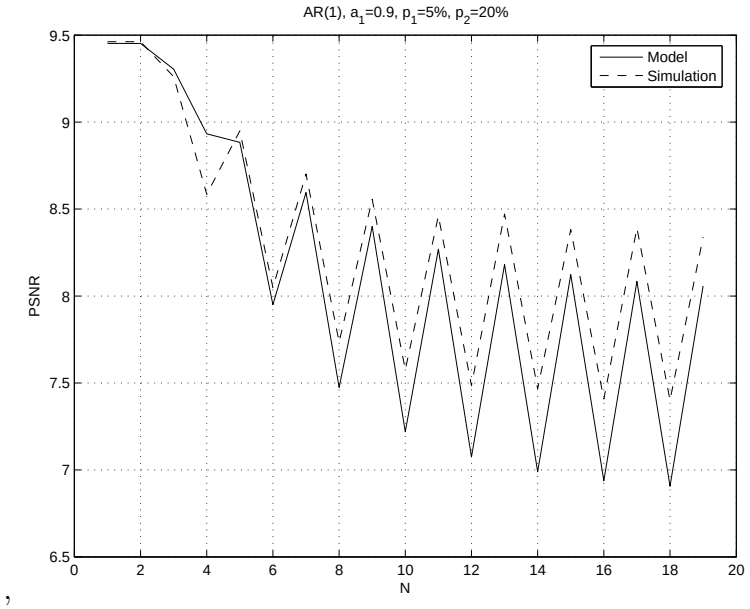
7.2 Extension of the Model for Video

In this section, we extend our R-Dd model for video sequences. The corresponding blocks from each frame along the sequence are considered as samples of an AR(1) source while each block in a frame belongs to a different AR(1) source. Recalling our model in section 7.1 the prediction coefficient between corresponding blocks is $a_2 = 1$. As in section 7.1, each frame is classified as odd or even numbered and sent on path 1 or path 2 based on this strategy. Each second corresponding block is sent on the same path encountering the same error probability. But before treating the blocks as one



,

Figure 7.1: One dimensional AR(1), $p_1 = 0\%, p_2 = 20\%$



,

Figure 7.2: One dimensional AR(1), $p_1 = 5\%, p_2 = 20\%$

dimensional-samples we recall that the pixels in a block are bilinearly interpolated to generate subpixel accuracy for motion estimation/compensation. We assume that the interpolation filter is the only loop filter and will use quarter pixel accuracy for our analysis. The analysis for half pixel or one eighth pixel accuracy is performed similarly. Assuming that the main source of error (besides quantization error) on each pixel is due to error concealment through motion based frame interpolation, we define a constant α_{interp} which gives the attenuation of the propagated distortion due to the subpixel motion compensation (loop filter). In section 7.2.1, we show why α_{interp} can be approximated as a constant over time for a block in a frame while each block has a different α_{interp} depending on its horizontal and vertical correlation coefficients ρ_h and ρ_v respectively. Below we will discuss the four identical reconstruction scenarios investigated in section 7.1 where x is the sequence of corresponding blocks and n is the frame number the block belongs to. $D(n)$ gives the distortion of block $x(n)$ in frame n whereas $D(n-2)$ gives the distortion of its corresponding block $x(n-2)$ in frame $n-2$.

Estimation of Distortion if blocks x(n) and x(n-2) are both received:

$$D(n) = \alpha_{interp}(D(n-2) - \sigma_{q2}^2) + \sigma_{q2}^2 \tag{7.10}$$

The residual error $e(n-2)-q_2(n-2)$ is filtered by the loop filter which attenuates its energy by α_{interp}.

Estimation of Distortion if block x(n) is received but x(n-2) is lost:

$$D(n) = \alpha_{interp}(D(n-2) + \sigma_{q2}^2) + \sigma_{q2}^2 \tag{7.11}$$

Estimation of Distortion if block x(n) is lost but x(n+1) is received:

$$D(n) = \frac{(1+\sqrt{\alpha_{interp}})^2}{4} D(n-1) + \frac{1+\alpha_{interp}}{4}\sigma_{q1}^2 \tag{7.12}$$

$$-\frac{(1+\sqrt{\alpha_{interp}})\sqrt{\alpha_{interp}}}{2}\sigma_{q1}^2(1-p_1) + \sigma_{interp}^2 \tag{7.13}$$

The interpolation error variance σ_{interp}^2 is different for each block of each frame.

To obtain the results presented in section 7.3, we interpolated (motion controlled) the lost frames using the previous and the next frames. The interpolated frame is subtracted from the original one yielding the interpolation error. We assume all corresponding blocks have the same interpolation error by using the above formula in each recursion with the same σ^2_{interp}. This is in practice not quite correct and depends on the motion vector fields as well as on the temporal correlation coefficient ρ_t of corresponding blocks.

Estimation of Distortion if blocks x(n) and x(n+1) are both lost:

$$D(n) = D(n-1) + \sigma^2_{rep} \tag{7.14}$$

The repetition error variance σ^2_{rep} is different for each block of each frame. To obtain results depicted in section 7.3, the repetition error is calculated by subtracting the original frame from the preceding one. The repetition error variance can then be calculated for each individual block. We assume by using the formula recursively with the same σ^2_{rep}, that the repetition for each corresponding block using the corresponding reference blocks will yield the same repetition error, which is not correct. As p_1, p_2 and N, the length of the video sequence increases, this last part of the equation produces smaller values than the real values occurring during the real transmission and decoding. To avoid this, we should consider a longer history by extending the last term of the formula. This point will be considered in further work.

7.2.1 Estimation of Error Propagation in Multi-Stream Video Transmission

We will mainly discuss four different cases and analyze how the interpolation error (the main source of error in multi-state recovery) is affected by the loop filter which performs nothing but subpixel interpolation. Denoting the errors on neighboring pixels (horizontal or vertical) on pixel grid by $e_1(n-1)$ and $e_2(n-1)$ with variance $D(n-1)$ over all pixels in the block, the error on possible four positions on the subpixel grid is calculated as listed below. According to the position x we analyzed the resulting error variance in three classes. Positions 2 and 4 are affected in the same way and therefore analyzed in the same class. The attenuation is analyzed

separately in horizontal and vertical directions. To ease the analysis, we assume bilinear interpolation in horizontal and vertical directions for all positions. $e_x(n-1)$ will be denoted as e_x in the sequel. $\alpha_{1,interp}$ is the attenuation in one direction whereas α_{interp} is the combined attenuation in both directions (product of one-dimensional attenuations).

Case 1, x=1

Only the pixels belonging to the grid of the original image are used at motion compensation, i.e. $\alpha_{1,interp} = 1$. The error variance $D(n-1)$ is not changed for this block.

Case 2, x=2,4 The subpixel used in motion compensation is one quarter pixel away from one of the original pixels and three quarters from the next one.

$$\begin{aligned} e_x &= \frac{3}{4}e_1 + \frac{1}{4}e_2 \\ \alpha_{1,interp} &= \frac{\sigma_{e,x}^2}{\sigma_{e,1}^2} \\ &= \frac{9}{16} + \frac{1}{16} + \frac{6}{16}\rho \\ &= \frac{5}{8} + \frac{3}{8}\rho \end{aligned}$$

where ρ is the horizontal or vertical correlation coefficient between adjacent error terms.

Case 3, x=3 The subpixel is in the middle of two original pixels.

$$\begin{aligned} e_x &= \frac{1}{2}e_1 + \frac{1}{2}e_2 \\ \alpha_{1,interp} &= \frac{1}{2} + \frac{1}{2}\rho \end{aligned}$$

Which of the above cases will be picked up, depends on the particular motion vector in horizontal and vertical direction. We assume that all 16 cases (all combinations of 4 horizontal and 4 vertical positions) occur with the same probability. According to that, the expected value of α_{interp} is calculated as follows:

$$\begin{aligned} E[\alpha_{interp}] &= \frac{1}{16}(1 + 2[\frac{5}{8} + \frac{3}{8}\rho_h] + [\frac{1}{2} + \frac{1}{2}\rho_h])(1 + 2[\frac{5}{8} + \frac{3}{8}\rho_v] + [\frac{1}{2} + \frac{1}{2}\rho_v]) \\ &= \frac{1}{256}(121 + 55\rho_h + 55\rho_v + 25\rho_h\rho_v) \end{aligned} \tag{7.15}$$

Since blocks in a frame have different ρ_h and ρ_v values, $E[\alpha_{interp}]$ should be calculated separately for each. $E[\alpha_{interp}]$ will be denoted by α_{interp} in the sequel. Since the error terms on blocks are assumed to be the accumulation of several error terms, the total error energy will be attenuated by the same value. In subsection 7.2.2, we show that the interpolation error on a block has the same ρ_h and ρ_v values as the block. The error terms constituted just by quantization are attenuated by $E[\alpha_{interp}] = \frac{121}{256} = 0.4727$ by setting $\rho_h = \rho_v = 0$ in Equation 7.15.

7.2.2 Analysis of the Interpolation Error

In subsection 7.2.1, we assumed that adjacent interpolation error values e_1 and e_2 are correlated with the same correlation coefficients ρ_h and ρ_v as the block pixels themselves. Here we will show that this assumption holds. The total distortion accumulated in each block is the sum of distortions due to previous interpolations. In the following c_x is the original sample which is to be reconstructed whereas a_x and b_x are the corresponding pixels from the previous and the next frames. x and $x-1$ denote the neighboring pixel positions in the block.

$$\begin{aligned} \sigma_{e,x}^2 &= R_{e,x}(0) \\ &= (c_x - \frac{a_x}{2} - \frac{b_x}{2})^2 \\ &= 1 + \frac{1}{4} + \frac{1}{4} - \rho_t - \rho_t + \frac{\rho_t^2}{2} \\ &= \frac{(\rho_t - 3)(\rho_t - 1)}{2} \end{aligned} \tag{7.16}$$

where ρ_t is the temporal correlation coefficient between corresponding blocks in adjacent frames. Similarly we calculate

$$\begin{aligned} R_{e,x}(1) &= E[(c_x - \frac{a_x}{2} - \frac{b_x}{2})(c_{x-1} - \frac{a_{x-1}}{2} - \frac{b_{x-1}}{2})] \\ &= \rho - \frac{\rho\rho_t}{2} - \frac{\rho\rho_t}{2} - \frac{\rho\rho_t}{2} + \frac{\rho}{4} + \frac{\rho_t^2\rho}{4} - \frac{\rho\rho_t}{2} + \frac{\rho_t^2\rho}{4} + \frac{\rho}{4} \\ &= \frac{\rho(\rho_t - 3)(\rho_t - 1)}{2} \end{aligned} \tag{7.17}$$

Combining Equations 7.16 and 7.17 we obtain:

$$\begin{aligned} \rho_{e,x} &= \frac{R_{e,x}(1)}{R_{e,x}(0)} \\ &= \rho \end{aligned} \tag{7.18}$$

ρ is the horizontal or the vertical correlation coefficient, the analysis can be performed separately in both directions.

7.3 Experiments

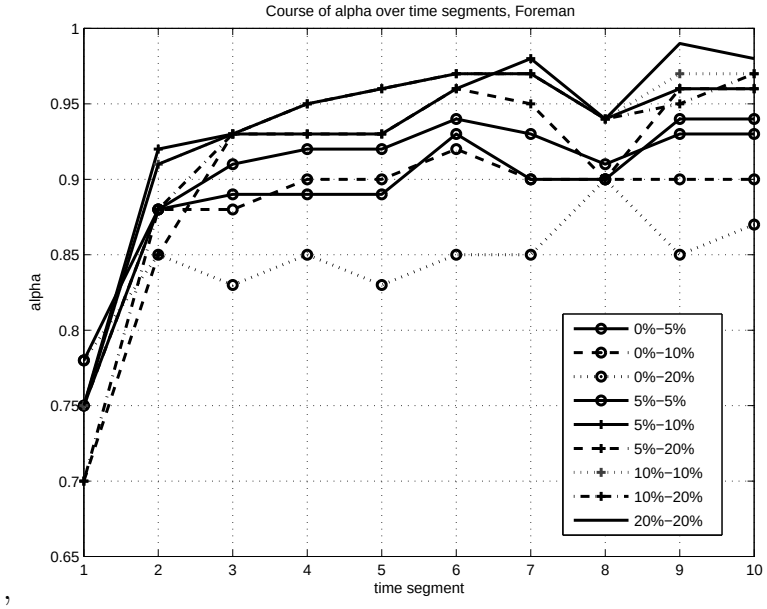
Figures 7.3, 7.4, 7.5, 7.6 and 7.7 show the course of α over time segments for the sequences Foreman, Akiyo, Claire, Mother & Daughter and Salesman respectively. Different from the single layered and temporal layered cases, α values are chosen to be dependent on the loss rates p_1 and p_2. If $p_1 = p_2$, as the loss rates increase α also increases. But if $p_1 < p_2$ as p_2 increases α decreases. This is the information gained from the five graphics. Besides, the course of α is also dependent on the sequence. This can be seen from the breaks of the curves at the same time points. Another observation is that in sequences with a low spatial and temporal activity, changes of α is restricted into a small range depending on the time segment as well as on the loss rates.

Figures 7.8, 7.9, 7.10, 7.11 and 7.12 show frame PSNR's plotted over the frame number N when $p_1 = 0\%$ and p_2 varies between 5% and 20% respectively. As seen the model estimation errors are smaller than 0.5 dB in each case.

Figures 7.11 and 7.12 show the accordance of the model estimation with the simulation results when both channels are lossy. Whereas Figure 7.11 belongs to balanced lossy channels Figure 7.12 shows the unbalanced lossy case.

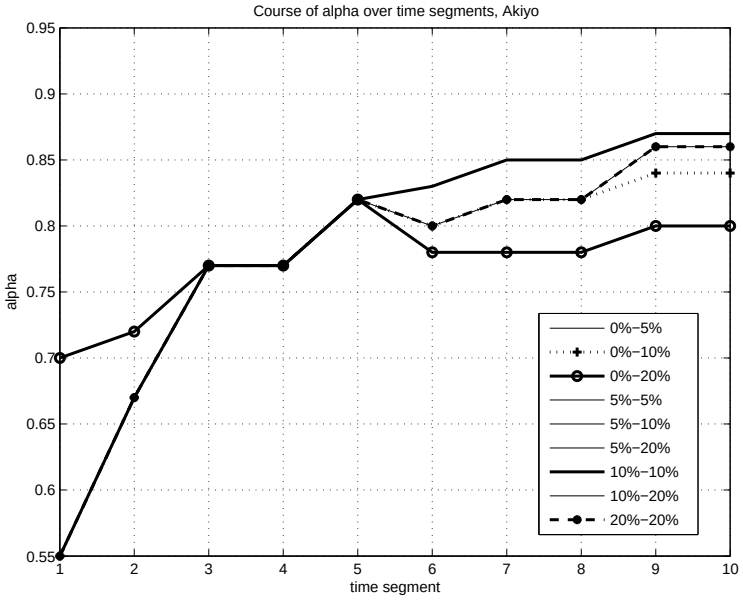
The next three Figures 7.13, 7.14 and 7.15 show the model-simulation accordance for Akiyo over 100 frames when $p_1 = 0\%$ and p_2 varies between 5% and 20%.

Figures from 7.16 till 7.23 show again the accordance of the model estimation with the simulation results for Akiyo, Claire, Mother & Daughter and Salesman when both channels are lossy. Figures 7.16, 7.18, 7.20 and 7.22 belong to the balanced lossy channels case whereas Figures 7.17, 7.19, 7.21 and 7.23 show the unbalanced lossy channel case.



,

Figure 7.3: MSVC, Course of α over time segments, Foreman



,

Figure 7.4: MSVC, Course of α over time segments, Akiyo

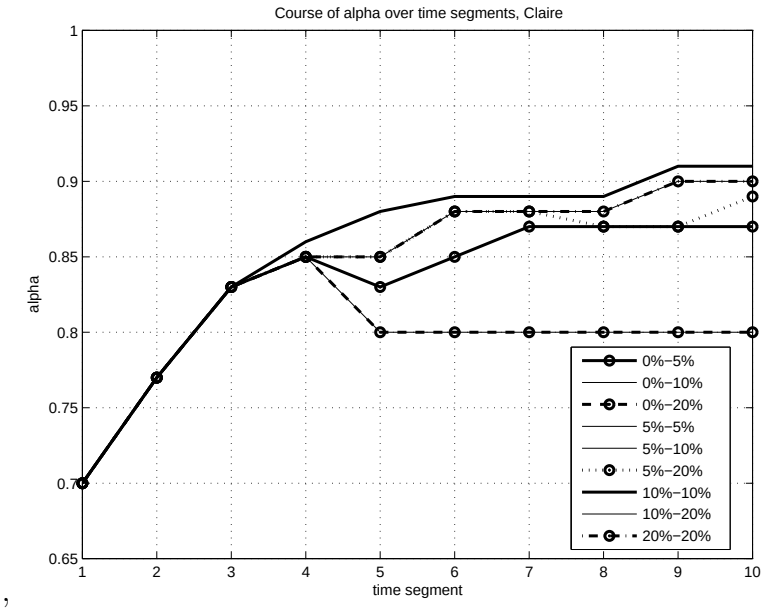


Figure 7.5: MSVC, Course of α over time segments, Claire

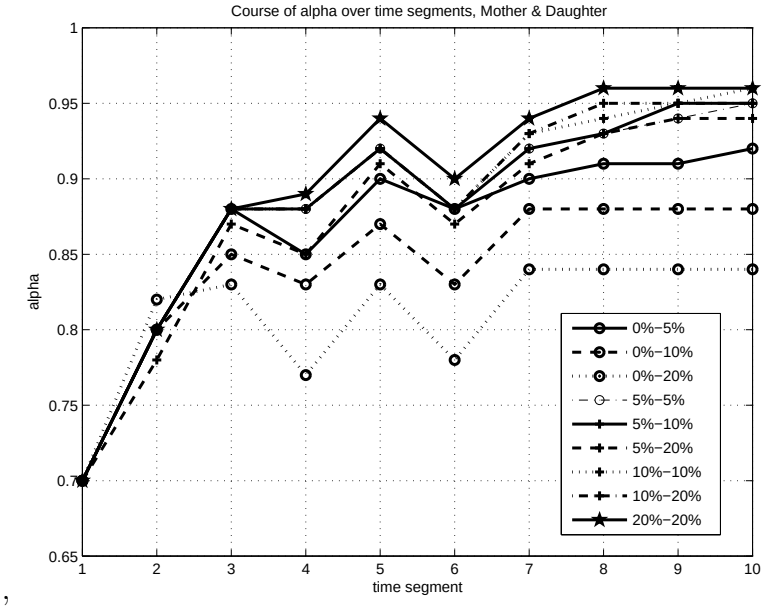


Figure 7.6: MSVC, Course of α over time segments, Mother & Daughter

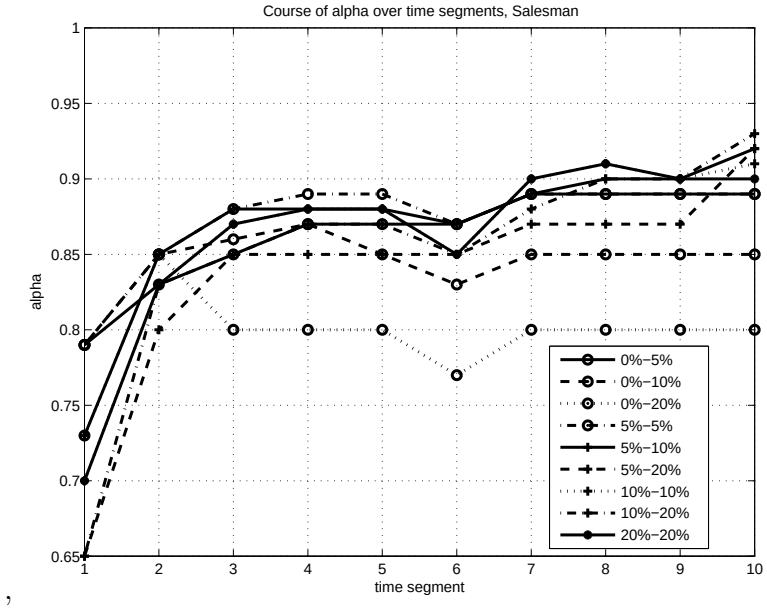


Figure 7.7: MSVC, Course of α over time segments, Salesman

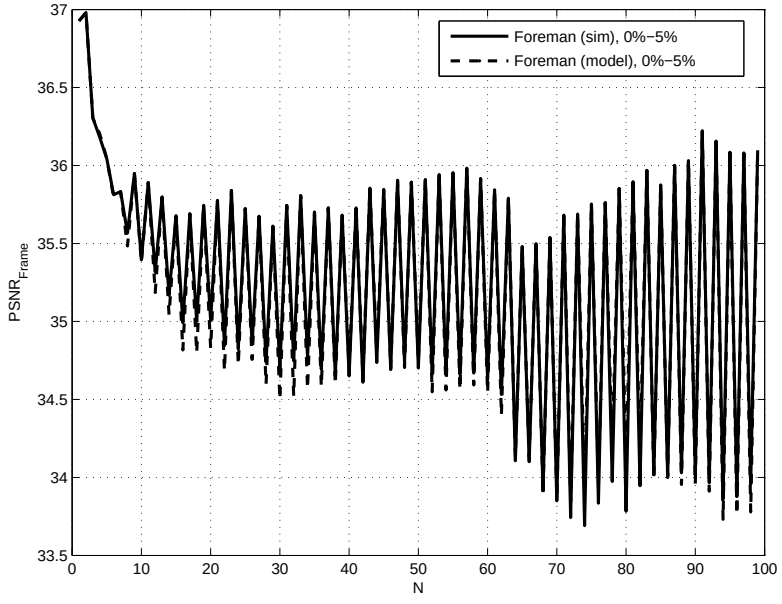


Figure 7.8: MSVC, Simulation vs. Model: Foreman, $p_1 = 0\%, p_2 = 5\%$

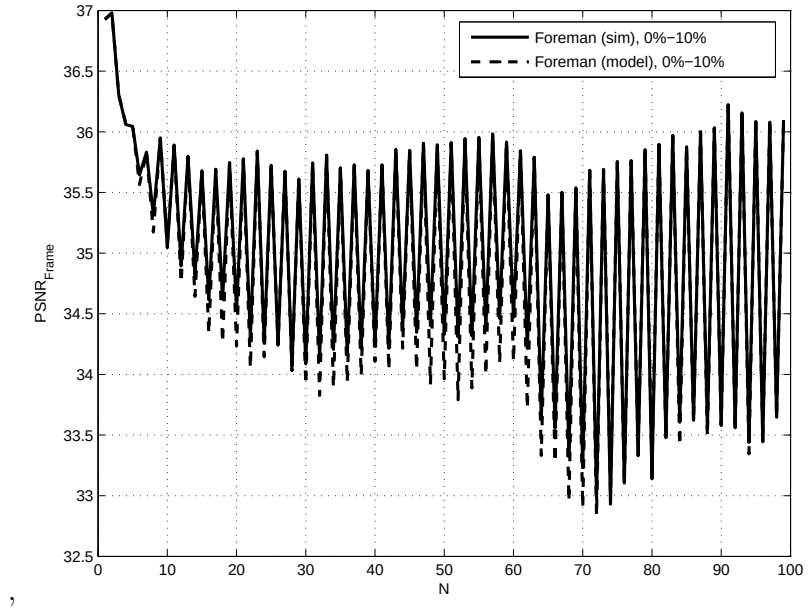


Figure 7.9: MSVC, Simulation vs. Model: Foreman, $p_1 = 0\%, p_2 = 10\%$

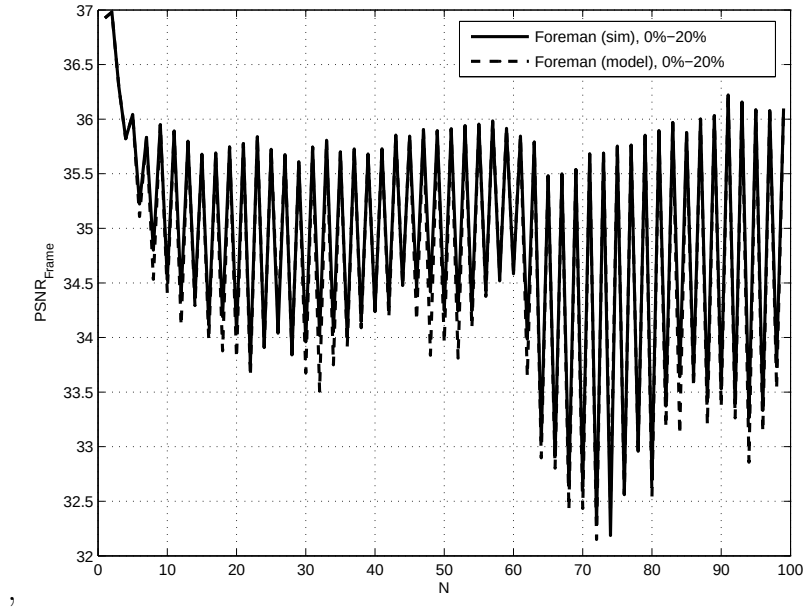


Figure 7.10: MSVC, Simulation vs. Model: Foreman, $p_1 = 0\%, p_2 = 20\%$

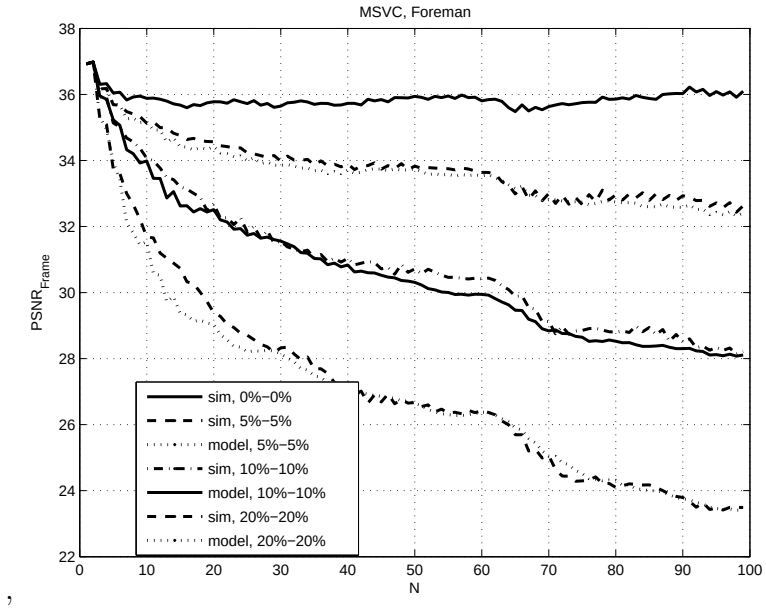


Figure 7.11: MSVC, Simulation vs. Model: Foreman, balanced loss probabilities

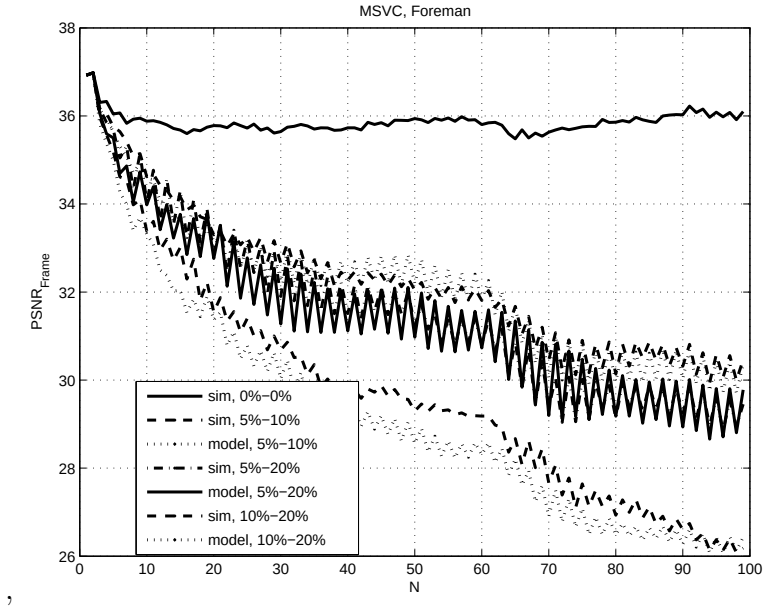
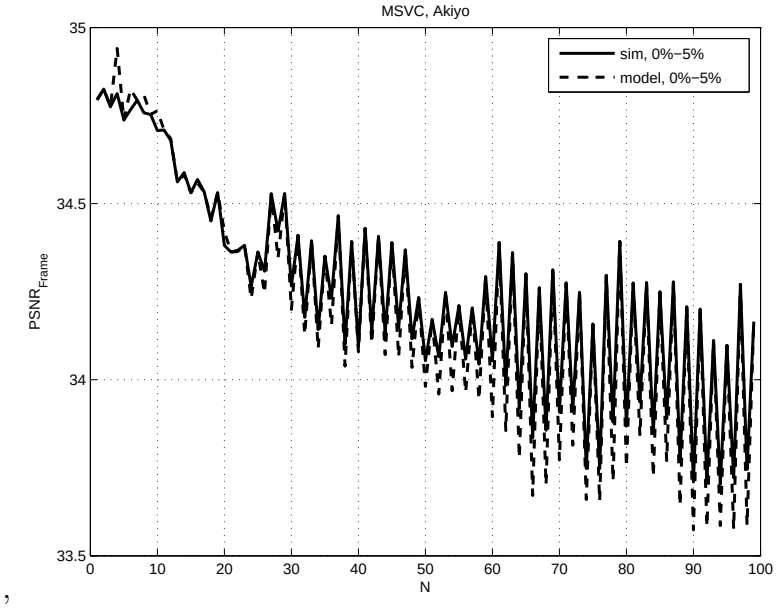
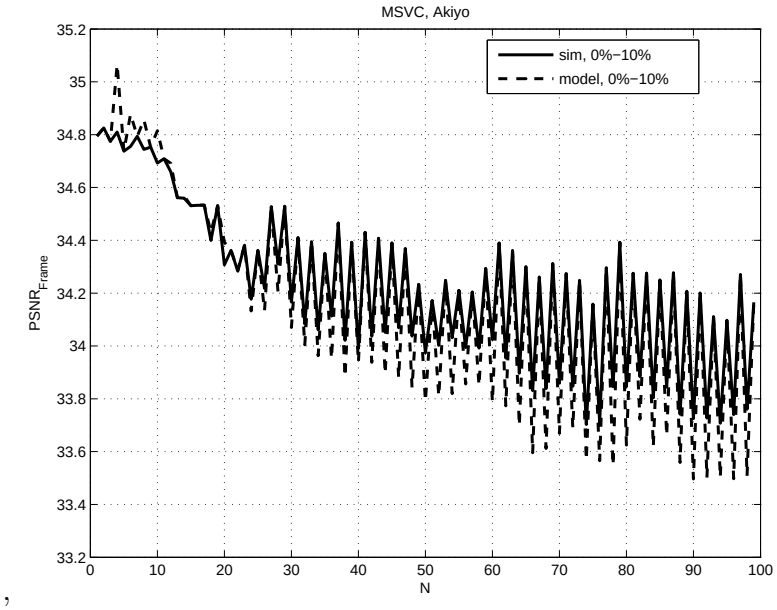


Figure 7.12: MSVC, Simulation vs. Model: Foreman, unbalanced loss probabilities



,

Figure 7.13: MSVC, Simulation vs. Model: Akiyo, $p_1 = 0\%, p_2 = 5\%$



,

Figure 7.14: MSVC, Simulation vs. Model: Akiyo, $p_1 = 0\%, p_2 = 10\%$

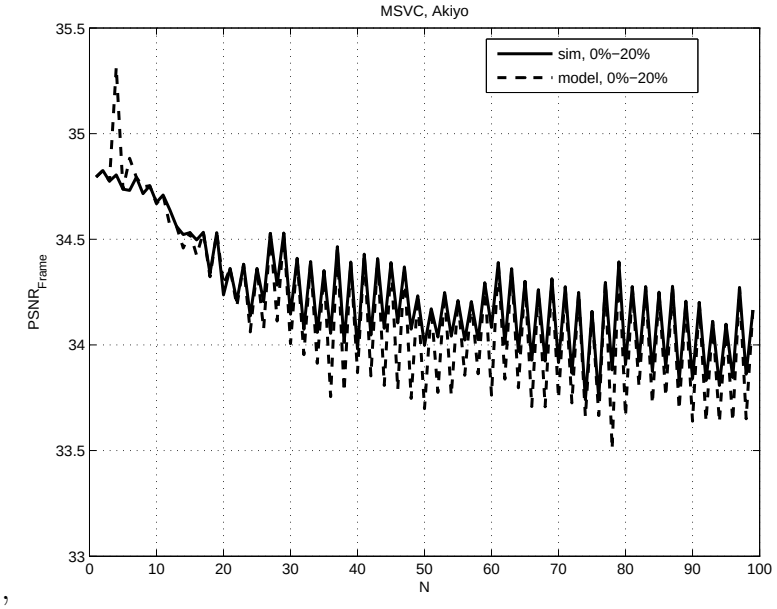


Figure 7.15: MSVC, Simulation vs. Model: Akiyo, $p_1 = 0\%, p_2 = 20\%$

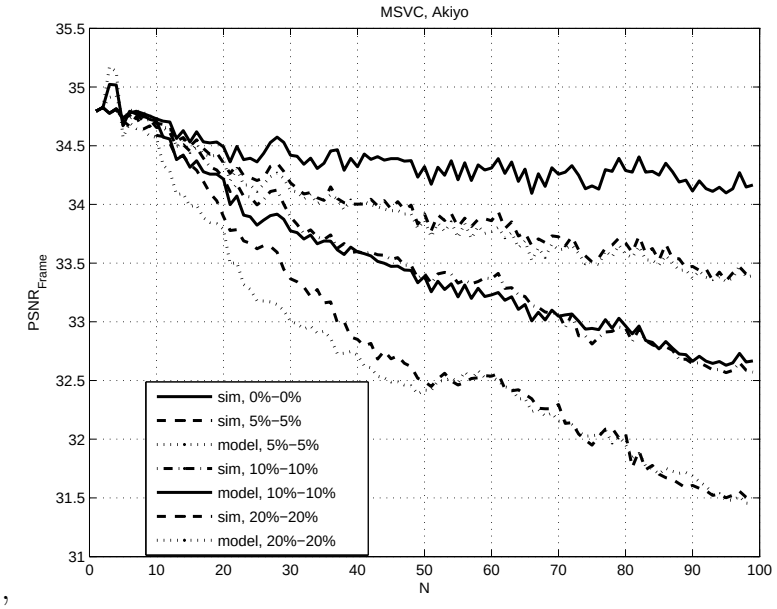
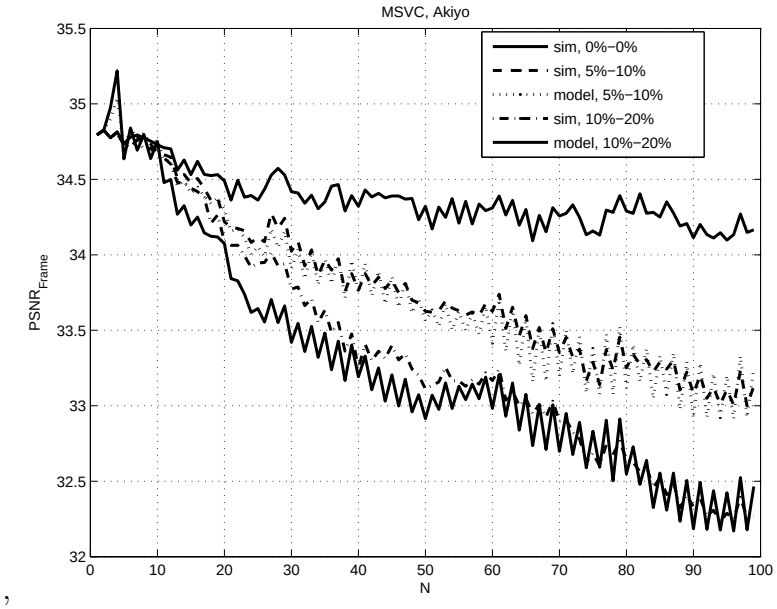
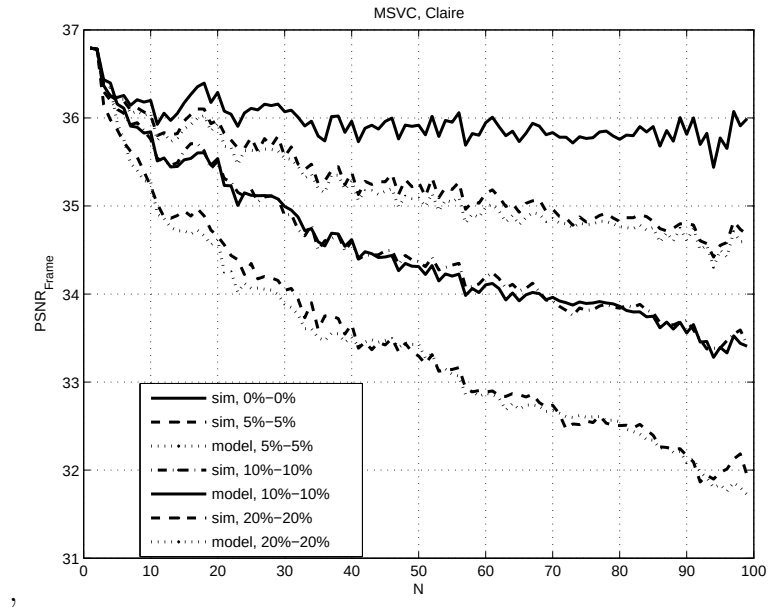


Figure 7.16: MSVC, Simulation vs. Model: Akiyo, balanced loss probabilities



,

Figure 7.17: MSVC, Simulation vs. Model: Akiyo, unbalanced loss probabilities



,

Figure 7.18: MSVC, Simulation vs. Model: Claire, balanced loss probabilities

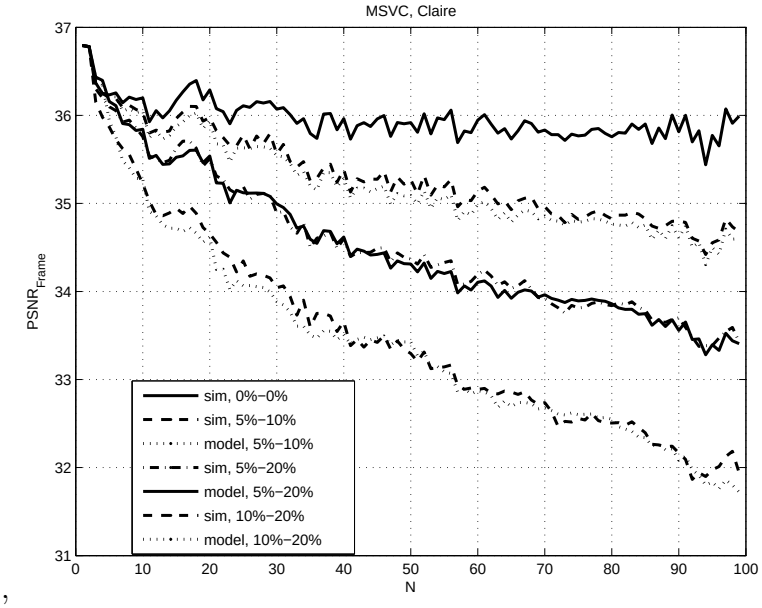


Figure 7.19: MSVC, Simulation vs. Model: Claire, unbalanced loss probabilities

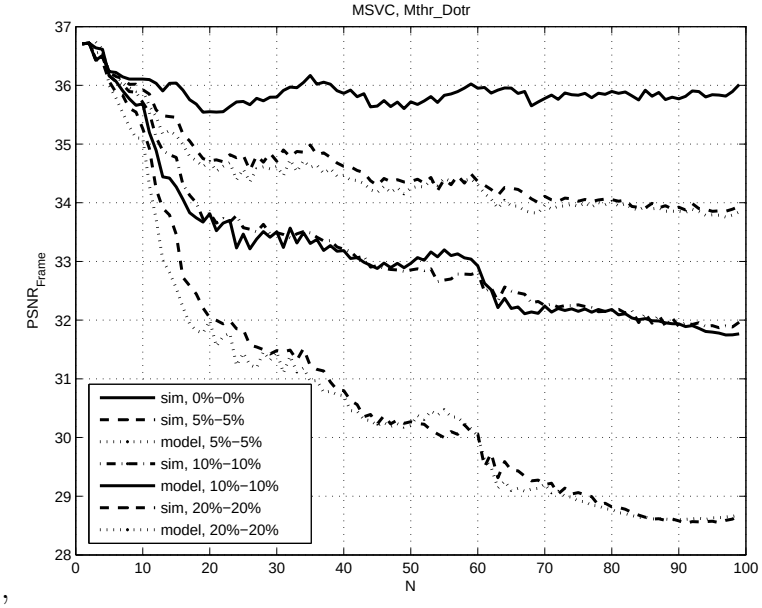
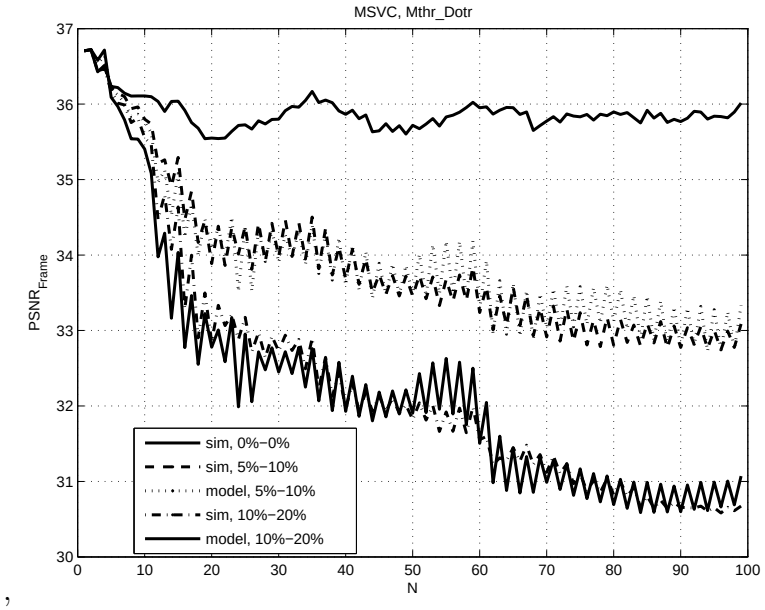


Figure 7.20: MSVC, Simulation vs. Model: Mother & Daughter, balanced loss probabilities



,

Figure 7.21: MSVC, Simulation vs. Model: Mother & Daughter, unbalanced loss probabilities

As stated from the comparison of the model results to the simulation results, the presented estimation technique yields very accurate results with a moderately low complexity. It is applicable to different coding options (e.g. intra/inter) and is extendable from Single State Video Coding to Temporal Layered and Multi-State Video Coding Systems. Moreover, it is an interesting technique for adaptive video streaming applications, where between different coding modes and coding methods can be switched depending on the estimated channel conditions like loss rate, bandwidth or delay. Moreover the estimation technique can also be employed to classify the video sequences or segments according to their motion characteristics. Another advantage of the technique is that it is not restricted to specific loss patterns or coding modes as compared to [133] and [169] respectively, and it can be applied for all kind of loss patterns and coding parameters.

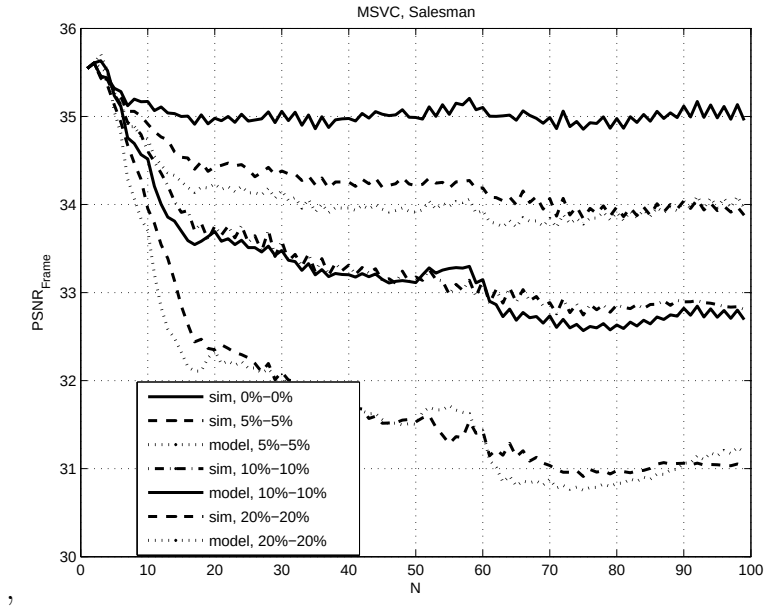


Figure 7.22: MSVC, Simulation vs. Model: Salesman, balanced loss probabilities

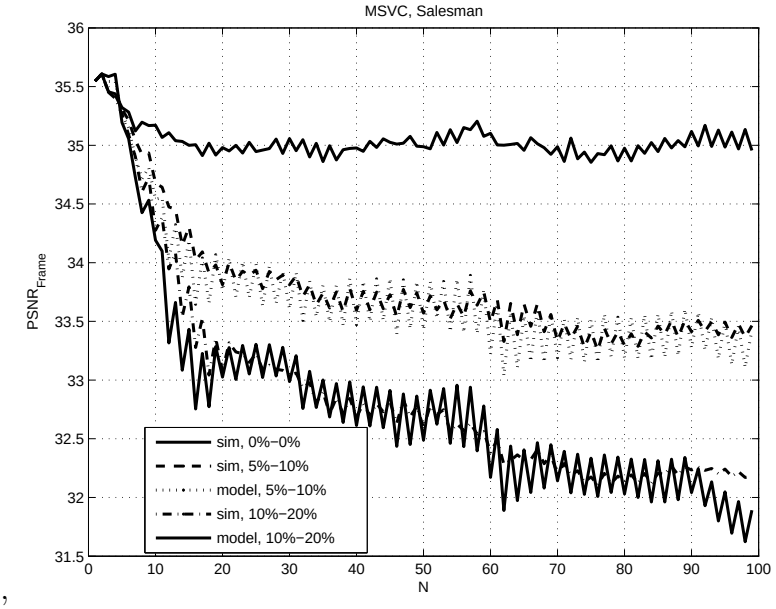


Figure 7.23: MSVC, Simulation vs. Model: Salesman, unbalanced loss probabilities

Chapter 8

Achieving Path Diversity

The benefits of path diversity are: 1-The application sees a virtual average path with a smaller variability in communication quality than over an individual path. 2-Burst packet losses are converted to isolated packet losses. 3-The probability of an outage is greatly reduced. During an outage all packets in the packet stream are lost. These improvements can be exploited to simplify design of packet-based communication systems. Two architectures are proposed in [4] for sending a stream of packets over multiple paths. The first one is based on IP Source routing whereas the second one is related to a relay infrastructure. The relay infrastructure is considered as an application-specific overlay network on top of the conventional Internet. It routes traffic through semi intelligent nodes at strategic locations in the Internet and provides a more reliable service over the Internet.

The main goal of employing path diversity is to improve communication over lossy packet networks. Video communication should be both effective and efficient. Since burst errors and outages are especially harmful to video communication, their effect is to be reduced as much as possible. But because of the dynamic nature of the packet networks such as Internet, it is difficult to fulfill these goals.

In the conventional Internet scenario, packets are dropped on the network with a destination IP address and the packets are to be delivered to that address. The sending and receiving end points have no control over the packets delivered from sender to the receiver. While one node or path in the network may be congested, other paths or nodes may have available bandwidths. One way to make use of this, is probing or listening to the channels to send packets along paths with available

bandwidth. However this method may be difficult since the congested areas can vary quite rapidly. But by using a number of paths at the same time the application can effectively see the average path behavior.

Several diversity techniques are studied in the context of wireless communication so far, such as frequency, time and spatial diversity. On the other hand, path diversity over wired packet networks is not widely explored. There exists great variability in the end-to-end performance observed over the Internet as discussed in [108]. The analogy of this variability to the variability existing in a wireless link motivates the use of diversity also for communication over the Internet. In [2] the performance of the default path between two hosts on the Internet is compared to the performance of alternative paths between them. According to this study, in 30-80% of the cases there is an alternate path with significant superior quality. Quality in this context is measured in terms of round-trip-time, loss rate and bandwidth. Routing diversity increases the applicability of MD coding, as does the high packet loss probability on wireless links [50], [51].

For achieving path diversity between points A and B, different packets are explicitly sent over different paths, e.g. half of the packets are sent over one path whereas the other half over another. Whereas in the ideal case, the number of transmitted packets remains constant, it may be required or be beneficial to send a larger total number of packets over multiple packets as compared to over the conventional single path. There are some important issues for path diversity systems to be considered: 1-the way to send packets through different paths, 2-the way to judge the degree to which two paths are different 3-the number paths to use 4-the way to share the load among the chosen paths and 5-the way to use feedback from the receiver to improve the communication.

The path taken by a packet depends on its starting address, destination address and the state of each of the routing tables for all of the routers traversed. The decision which next hop router to send the packet to is made by the current visited router. Routing the packet is the job of the network layer and this is transparent to the application. An important point is that the consecutive packets between a sender and receiver go through the same path, since the time rate of change of routing table is slow as compared to the time between the packets. Although consecutive packets go through the same path, they may see different delay and loss characteristics. The

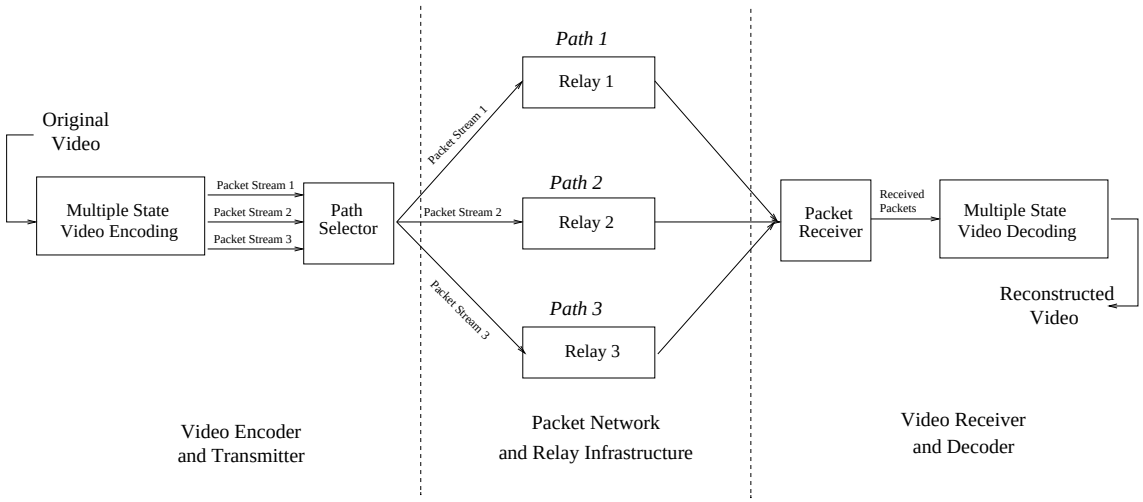


Figure 8.1: A Path Diversity System

cross traffic at a node can vary quite quickly changing the congestion as well as the delay and loss characteristics. Figure 8.1 shows an example of a path diversity system. Path Diversity via IP Source Routing is discussed in 8.1 and Path Diversity via Relays in 8.2. Benefits of using a path diversity system is given in section 8.3.

8.1 Path Diversity via IP Source Routing

In "source routing", one explicitly specifies the set of nodes for each packet to transverse. In loose source routing a subset of the nodes is specified, in strict source routing on the other hand, the complete set of nodes is specified. Path diversity can be achieved by specifying different source routes for different subsets of packets.

Although IP Source Routing sounds straightforward it has a lot of problems. The first one is the security problem arising from allowing any end user to specify any source routes. IP Source Routing is not turned on within the Internet because of security. It may be allowed within a private network . Even in this case there is the problem of choosing the appropriate paths. Here the knowledge of the network topology and the network state is needed which is a complicated issue. One needs to know node addresses within the various subnetworks that must be crossed to specify a route through the Internet. However the topology of the Internet is ever changing and it would not be a good idea to design an algorithm that requires detailed knowledge

of the network topology. Secondly, it is impractical to know the network stage for a large part of the internet. This might be still possible for a company network which is controlled and tractable. Moreover, more overhead is required to carry the routing information since the intermediate node addresses must be included in each packet. To sum up, IP source routing is probably not appropriate for use over the Internet while being attractive for a much smaller company network.

8.2 Path Diversity via Relays

Another approach for path diversity is through the use of relays. They are replaced in a number of important nodes in the infrastructure and each of them performs a simple forwarding operation. Here an original packet having a destination address and a payload is encapsulated into a new packet which is sent to a relay. The relay receives the packet, strips of its address and drops the original packet back on the network where it goes off to the final destination. Suppose sender A wants to transmit a stream of packets to receiver B and relay C is available. Sender A can send a subset of the packets directly to B by dropping them on the network as usual and the rest of the packets can be send to C after encapsulating them in a new packet. C forwards the received packets further to B. There are more alternatives for achieving path diversity if there are more available relays. Sometimes it may be necessary to use separate relays to ensure separate paths and sometimes not. Alternatively, the relay may examine the destination address and perform some appropriate processing increasing the complexity of the system. The architecture itself provides freedom to optimize the relay network. Most of the freedom may be hidden from the applications. There are a number of ways in which an application can interact with a relay network to provide path diversity. First alternative is just knowing the IP address of an appropriate relay and communicating directly with it. The second alternative is, if a relay network architecture or service is established, sending the packets to the service so that the service automatically provides the relaying and ensures reliable delivery. Figure 8.2 shows the block diagram of a path diversity system based on a relay network.

To sum up, IP Source Routing is a straightforward method to send different packets over different paths. Although theoretically simple, it is practically impossible

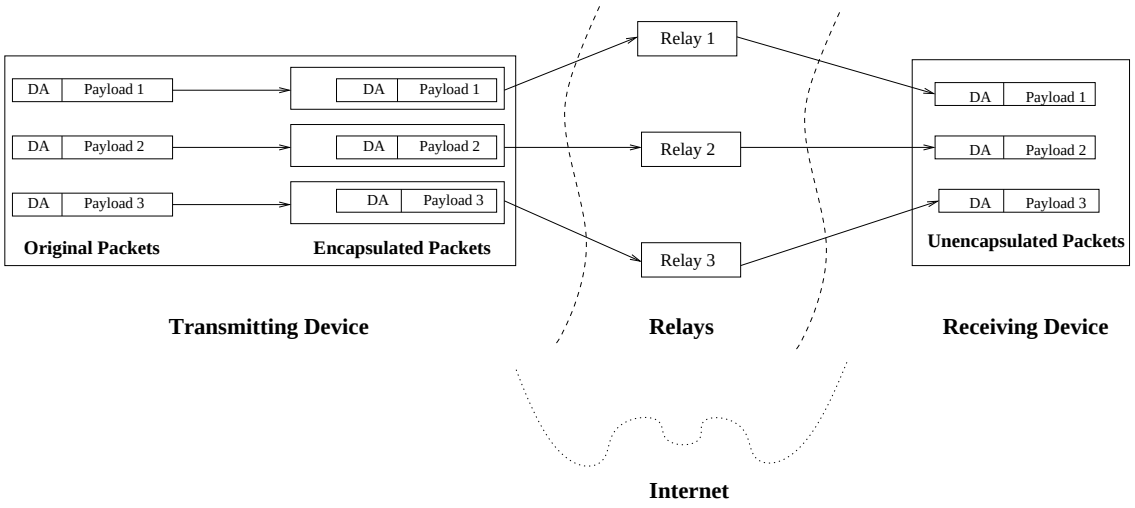


Figure 8.2: Path diversity through the use of a relay network

over the internet. Source routing is turned off because of the security reasons and IP source routing may only be usable within a company intranet. Relay based path diversity on the other hand, provides simple method for achieving path diversity. The difficulty is the requirement of an infrastructure of relays.

8.3 Benefits of Path Diversity

Explicitly sending a packet stream over multiple paths provides some advantages, which are:

1- The average network behavior has less variability in link/application quality. A reduced variability, on the other hand eases the system design and improves the end-to-end application quality.

2- Loss of many consecutive packets (burst loss) can be converted into the loss of a number of isolated packets. This helps, since it is much easier to recover from multiple isolated losses than from a number of consecutive losses.

3- The probability of an outage decreases with path diversity. An outage occurs only if and only if all paths undergo outages simultaneously. If the loss processes of each path can be modeled as independent and identically distributed and if the outage probability of each is P_{outage} the outage probability over multiple paths is P_{outage}^{N} where N is the number of paths. For all applications with delay constraints or

for which retransmissions are not possible, a reduction in outage probability is a big advantage.

Chapter 9

Conclusion

The thesis is focused on exploring the performance of Multi-State Video Coding (MSVC) proposed first in [4] as a Multiple Description Scheme for Video Coding. The investigation is performed through simulations as well as through a theoretical framework based on decoder distortion estimation in lossy transmission. The performance is measured in terms of average peak signal to noise ratio of the reconstructed video sequence in a lossy environment where the generated substreams using MSVC are transmitted over different paths to the receiver. The channels are considered in terms of their packet loss probabilities and the simulation results are obtained by averaging the frame PSNR's generated using different loss patterns at the same loss probabilities of the channels.

In addition to the original MSVC scheme which we called Approach 1, an improved version, Approach 2 is developed which is capable of giving a gain of up to 1 dB over Approach 1 depending on the unbalance in rate allocation of the two streams, on the sequence and also on the amount and type of intra coding incorporated to the system. Moreover the trade-off between the amount of intra coding and the quantization stepsize is explored in terms of its effect on the system performance while the total encoding bitrate is kept constant. It is seen that in case of high motion sequences, it is useful to combine MSVC with frame intra-updates whereas for low motion sequences at low and moderately high loss probabilities no updates are necessary. State recovery works good enough to keep the system performance at an acceptable level. Additionally, frame intra-updates work more efficient than intra-GOB updates.

Next we compared MSVC to SSVC, (Single State Video Coding), and to TLC (Temporal Layered Coding). We assumed that both MSVC and TLC use two channels for transmission, whereas SSVC employs just one channel. In the first case we assume that the motion vectors for TLC and SSVC are always received. In the second case, however, we consider that the motion information is also lost when the corresponding frame packet is lost. In the first case, both SSVC and TLC use motion compensated error concealment, where the blocks in the previous frame are moved to the positions pointed to by the motion vectors. On the other hand, in the second case, since no motion vectors are available, the previous frame is repeated to replace the current lost frame. The observation in the first case is if the motion vectors are always received for SSVC and TLC, they both outperform MSVC. The difference between SSVC and MSVC is even larger for low motion sequences since the penalty of using two streams instead of one and decreasing the prediction gain plays a much bigger role for low motion sequences due to the larger original prediction gain. On the other hand, if the motion vectors are lost, MSVC outperforms SSVC and TLC, especially for high motion sequences where the strength of state recovery becomes obvious as compared to frame repetition.

In the second part of the work the focus is on decoder distortion estimation given the sequence, the loss probability of the channel, and the quantization variance. The estimation algorithm models each block in a frame with its corresponding blocks along the sequence as an AR(1) source with a specific correlation coefficient. The correlation coefficient is dependent on the spatial correlation in the block. Although one may assume that the spatial correlation in the block will be the same for all the corresponding blocks in the sequence, it is much better to set or calculate this value adaptive to the sequence, e.g. for each segment of 10 frames. The decoder distortion can be estimated recursively for each block of each frame depending on its corresponding blocks located in the previous frames. The results obtained using the model are verified by a comparison with the simulation results. The simulation results are obtained using 100 different loss patterns for each loss rate. The sequences are decoded for the given loss patterns, applying the appropriate concealment mechanism and averaging the obtained frame PSNR's over all patterns. The model was first designed for SSVC transmission and then extended to estimate the decoder distortion for temporal layered (TLC) and multi-state video coding (MSVC). In MSVC case, the

algorithm should be extended to take the error concealment through state recovery into account. But still each block with its corresponding blocks along the sequence is considered as an AR(1) source. Different than SSVC and TLC, wee see that the correlation coefficient associated with each block changes with changing loss probabilities of the channels. A comparison of the estimated results with the simulation results verifies the model.

At last, we discuss the ways of achieving path diversity. Mainly two methods are recalled: Path Diversity via IP Source Routing and Path Diversity via Relays. In the rest of the work, we considered employing path diversity without concentrating on the method how to achieve it.

Appendix A

Acronyms

MSVC	Multi-State Video Coding
SSVC	Single-State Video Coding
SSVC_TC	Single-State Video Coding with Temporal Concealment
TLC	Temporal Layered Coding
TLC_TC	Temporal Layered Coding with Temporal Concealment
AR	The Autoregressive Process
MDC	Multiple Description Coding
PSNR	Peak Signal to Noise Ratio
GOB	Group of Blocks
TCP	Transmission Control Protocol
JPEG	Joint Photographic Experts Group
PCM	Pulse Code Modulation
DPCM	Differential Pulse Code Modulation
BCC	Binary Symmetric Source
SR	Successive Refinement
UEP	Unequal Error Protection
QFE	Quantized Frame Expansion
FEC	Forward Error Correction
RTT	Round Trip Time
VRC	Video Redundancy Coding
QCIF	Quarter Common Intermediate Format
ME	Motion Estimation

MC	Motion Compensation
RD	Rate Distortion
ARQ	Automatic Repeat Request
MB	Macroblock
mcdb	Motion Compensated Difference Block
mcdf	Motion Compensated Difference Frame
R-Dd	Rate Decoder Distortion

Appendix B

Symbols

$PSNR\_avg$	average peak signal to noise ratio
R_T	total bitrate
R_1	rate allocated to the first description
R_2	rate allocated to the second description
D	average distortion
D_1	distortion of the first stream
D_2	distortion of the second stream
N	number of descriptions
QP	quantization stepsize
p_1	loss rate of the first channel
p_2	loss rate of the second channel
x	source signal
b	block
$d = e$	total error on the block
q	quantization error on the block
F	frame
σ^2	variance
mv	motion vector of the block
ρ_1	normalized first order autocorrelation coefficient
σ^2_{interp}	interpolation error variance
σ^2_{rep}	repetition error variance
σ^2_q	quantization error variance

Curriculum Vitae

Personal Data:

Name:	Sila Ekmekci
Date of birth:	15. June 1973
Place of birth:	Frankfurt a.M., Germany
Nationality:	Turkish

Education:

09.1979 - 06.1984:	Muhittin Ustundag Ilkokulu, Elementary School Istanbul, Turkey
09.1984 - 06.1991:	Istanbul Lisesi, German Highschool Istanbul, Turkey
10.1991 - 06.1996:	Bogazici University Bachelor of Science in Computer Engineering, Istanbul, Turkey
10.1991 - 06.1996:	Bogazici University Bachelor of Science in Electrical Engineering, Istanbul, Turkey
10.1996 - 05.1998:	Technical University Berlin, Diploma (MS equivalent) in Electrical Engineering, Berlin, Germany
02.2000 - 05.2000:	Georgia Institute of Technology, Visiting Ph.D. student in Electrical Engineering, Atlanta, GA, USA
09.2001 - 04.2002:	Stanford University, Visiting Ph.D. student in Electrical Engineering, Palo Alto, CA, USA

08.1998 - 12.2004:	Technical University Berlin, Ph.D. in Electrical Engineering, Berlin, Germany

Occupation:

06.1997 - 10.1998:	Heinrich Herzt Institute, Researcher at Image Processing Department, Berlin, Germany
11.1998 - 06.2004:	Technical University Berlin, Research and Teaching Assistant at Telecommunications Institute, Berlin, Germany
07.2004 - current:	Swiss Institute of Technology, Postdoctoral Scholar at Signal Processing Institute, Lausanne, Switzerland

Bibliography

[1] A. Aaron, R. Zhang, and B. Girod. Wyner-ziv coding of motion video. *Proc. Asilomar Conference on Signals and Systems*, November 2002.

[2] A.C.Savage, S.E.Hoffman, J. Snell, and T. Anderson. The end-to-end effects of internet path selection. *Proc. of the ACM SIGCOMM*, October 1999.

[3] R. Ahlswede. On multiple descriptions and team quessing. *IEEE Trans. Inform. Theory, vol. 32, no. 4, pp. 543-549*, July 1986.

[4] J. Apostolopoulos. Reliable video communication over lossy packet networks using multiple state encoding and path diversity. *VCIP*, January 2001.

[5] J. Apostolopoulos and S.J. Wee. Unbalanced multiple description video communication using path diversity. *ICIP*, October 2001.

[6] R. Aravind, M. R. Civanlar, and A. R. Reibman. Packet loss resilience of MPEG-2 scalable video coding algorithms. *IEEE Trans. Circuits Syst. Video Technol, vol. 6, pp. 426-435*, October 1996.

[7] R. Arean, J. Kovacevic, and V.K. Goyal. Multiple description perceptual audio coding with correlating transforms. *IEEE Trans. Speech Audio Processing, vol. 8, no. 2, pp. 140-145*, March 2000.

[8] R. Balan, I. Daubechies, and V. Vaishampayan. The analysis and design of windowed fourier frame based multiple description source coding schemes. *IEEE Trans. Inform. Theory, vol. 46, no. 7, pp. 2491-2536*, November 2000.

[9] C. Batllo and V.A. Vaishampayan. Asymtotic performance of multiple description transform codes. *IEEE Trans. Inform. Theory, vol. 43, no. 2, pp. 703-707*, March 1997.

[10] T. Berger. Rate distortion theory. *Englewood Cliffs, NJ: Prentice-Hall*, 1971.

[11] T. Berger and Z. Zhang. Minimum breakdown degradation in binary source encoding. *IEEE Trans. Inform. Theory, vol. 29, no. 6, pp. 807-814*, November 1983.

[12] D.D. Botvich, T. Curran, A. MacFhearraigh, and S. Murphy. Hierarchical approach to video source modelling. *IEE*, 1994.

[13] J. M. Boyce and R. D. Gaglianello. Packet loss effects on MPEG video sent over the public internet. *ACM Multimedia 98*, September 1998.

[14] C.-S.Kim and S.-U. Lee. Multiple description motion coding algorithm for robust video transmission. *Proc. IEEE Int. Symp. Circuits and Systems, vol. 4, pp. 717-720*, May 2000.

[15] R. Calderbank, I. Daubechies, W. Sweldens, , and B.L.Yeo. Wavelet transforms that map integers to integers. *Appl. Comput. Harmon. Anal., vol. 5, no. 3, pp. 332-369*, July 1998.

[16] C. Candy, D. Gloge, D.J. Goodman, W.M. Hubbard, and K. Ogawa. Protection by diversity at reduced quality. *Memo for Record*, June 1978.

[17] C. Chen. *Motion-Compensated Hybrid Coders in Video Communications*. Ph.D. Thesis, Monash University, Australia, 1992.

[18] J. Choi and D. Park. A stable feedback control of the buffer state using the controlled lagrange multiplier method. *IEEE Trans. Image Processing, vol 3, pp, 546-588*, September 1994.

[19] P.A. Chou, S. Mehrotra, and A. Wang. Multiple description decoding of overcomplete expansions using projections onto convex sets. *Proc. IEEE Data Compression Conf., pp. 72-81*, March 1999.

[20] D.-M. Chung and Y. Wang. Multiple description image coding using signal decomposition and reconstruction based on lapped orthogonal transforms. *IEEE Trans. Circuits Syst. Video Technol., vol. 9, no. 6, pp. 895-908*, Sept. 1999.

[21] D. Comas, R. Singh, and A. Ortega. Rate-distortion optimization in a robust video transmission based on unbalanced multiple description coding. *2001 Workshop on Multimedia Signal Processing*, 2001.

[22] J.H. Conway and N.J.A. Sloane. Fast quantizing and decoding algorithms for lattice quantizers and codes. *IEEE Trans. Inform. Theory, vol. 28, no. 2, pp. 227-232*, March 1982.

[23] G. Cote and F. Kossentini. Optimal intra coding of blocks for robust video communication over the internet. *Image Communication, pp. 25-34*, September 1999.

[24] T.M. Cover and J.A. Thomas. Elements of information theory. *Wiley*, 1991.

[25] D.A.Patterson, G. Gibson, and R.H. Katz. A case for redundant arrays of inexpensive disks. *Proc. ACM SIGMOD, pp. 109-116*, June 1988.

[26] I. Daubechies. Ten lectures on wavelets. *SIAM*, 1992.

[27] S. Deering and R. Hinden. *Internet Protocol. version 6 (IPv6) specification. Network Working Group Request for Comments 1883*, December 1995.

[28] S.N. Diggavi, N.J.A. Sloane, and V.A. Vaishampayan. Design of asymmetric multiple description lattice vector quantizers. *Proc. IEEE Data Compression Conference*, March 2000.

[29] P.L. Dragotti, J. Kovacevic, and V.K. Goyal. Quantized oversampled filter banks with erasures. *Proc. IEEE Compression Conf., pp. 173-182*, March 2001.

[30] P.L. Dragotti, S.D. Servetto, and M. Vetterli. Analysis of optimal filter banks for multiple description coding. *Proc. IEEE Data Compression Conf., pp. 323-332*, March 2000.

[31] R.J. Duffin and A.C. Schaeffer. A class of nonharmonic Fourier series. *Trans. Amer. Math. Soc., vol 72, pp. 341-366*, 1952.

[32] M. Effros. Optimal modeling for complex system design. *IEEE Signal Processing Magazine, vol. 15, pp. 51-73*, November 1998.

[33] M. Effros and A. Goldsmith. Capacity definitions and coding strategies for general channels with receiver side information. *Proc. IEEE Int. Symp. Information Theory, p. 39*, August 1998.

[34] S. Ekmekci and T. Sikora. Unbalanced quantized multiple description video transmission using path diversity. *Electronic Imaging 2003, SPIE*, January 2003.

[35] S. Ekmekci and T. Sikora. Temporal layered vs. multistate video coding. *IS&T/SPIE's Electronic Imaging 2004*, January 2004.

[36] S. Ekmekci and T. Sikora. Model for unbalanced multiple description video transmission using path diversity. *VCIP 2003, SPIE*, July 2003.

[37] S. Ekmekci and T. Sikora. Multi-state vs. single-state video coding over error-prone channels. *2003 Asilomar Conference*, November 2003.

[38] W.H.R. Equitz and T.M. Cover. Addendum to successive refinement of information. *IEEE Trans. Inform. Theory, vol. 39, pp. 1465-1466*, July 1993.

[39] W.H.R. Equitz and T.M. Cover. Successive refinement of information. *IEEE Trans. Inform. Theory, vol. 37, pp. 269-275*, March 1991.

[40] T. Ericson and V. Ramamoorthy. Modulo-PCM: A new source coding scheme. *Proc. IEEE. Int. Conf. Acoustics, Speech and Signal Processing, pp. 419-422*, April 1979.

[41] M. Fleming and M. Effros. Generalized multiple description vector quantization. *Proc. IEEE Data Compression Conf., pp. 3-12*, March 1999.

[42] M. Flierl, T. Wiegand, and B. Girod. A locally optimal design algorithm for block-based multi-hypothesis motion-compensated prediction. *Proceedings of the IEEE DCC*, March 1998.

[43] Y. Frank-Dayan and R. Zamir. Universal lattice-based quantizers for multiple descriptions. *Proc. IEEE Data Compression Conf., pp. 500-509*, March 2000.

[44] R. Frederick, H. Schulzrinne, S. Casner, and V. Jacobson. RTP: a transport protocol for real-time applications. *RFC 1889, http://www.faqs.org/rfcs/rfc1889.html*, January 1996.

[45] A. A. El Gamal and T. M. Cover. Achievable rates for multiple descriptions. *IEEE Trans. Inform. Theory, vol. 28, pp. 851-857*, November 1982.

[46] A. Gersho. The channel splitting problem and Modulo-PCM coding. *Bell Labs Memo for Record (not archived)*, October 1979.

[47] A. Gersho and R.M. Gray. Vector quantization and signal compression. *Kluwer*, 1992.

[48] B. Girod and N. Faerber. Feedback-based error control for mobile video transmission. *IEEE, Special Issue on Video for Mobile Multimedia*, 1999.

[49] B. Girod, T. Wiegand, E. Steinbach, M. Flierl, and X. Zhang. High-order motion compensation for low bit-rate video. *Proceedings of the European Signal Processing Conference*, September 1998.

[50] N. Gogate, D-M. Chung, S.S. Panwar, and Y. Wang. Supporting image and video applications in a multihop radio environment using path diversity and multiple description coding. *IEEE Transactions on Circuits and Systems for Video Technology , Vol.12, Issue.9, pp.777-792.*, September 2002.

[51] N. Gogate and S.S. Panwar. Supporting image/video applications in a multihop radio environment using route diversity. *Proc. IEEE Int. Conf. Communications, vol. 3, pp. 1701-1706*, June 1999.

[52] V.K. Goyal. Transform coding with integer-to-integer transforms. *IEEE Trans. Inform. Theory, vol. 46, pp. 465-473*, March 2000.

[53] V.K. Goyal. Multiple description coding: Compression meets the network. *IEEE Signal Processing Mag., vol. 18, no. 5, pp. 74-93*, Sept. 2001.

[54] V.K. Goyal. Theoretical foundations of transform coding. *IEEE Signal Processing Mag., vol. 18, pp. 9-21*, September 2001.

[55] V.K. Goyal, J.A. Kelner, and J. Kovacevic. Multiple description vector quantization with a coarse lattice. *IEEE Trans. Inform. Theory, vol. 48, no. 3, pp. 781-788*, March 2002.

[56] V.K. Goyal and J. Kovacevic. Optimal transform coding of gaussian vectors. *Proc. IEEE Data Compression Conference, pp. 388-397*, March 1998.

[57] V.K. Goyal and J. Kovacevic. Generalized multiple description coding with correlating transforms. *IEEE Trans. Inform. Theory, vol.47*, September 2001.

[58] V.K. Goyal, J. Kovacevic, R. Arean, and M. Vetterli. Multiple description transform coding of images. *Proc. IEEE Int. Conf. Image Processing, vol. 1, pp. 674-678*, October 1998.

[59] V.K. Goyal, J. Kovacevic, and J.A. Kelner. Quantized frame expansions with erasures. *Applied Comput. Harmon. Anal., vol.10, no.3, pp. 203-233*, May 2001.

[60] V.K. Goyal, J. Kovacevic, and M.Vetterli. Multiple description transform coding: Robustness to erasures using tight frame expansions. *Proc. IEEE Int. Symp. Information Theory, p. 408*, August 1998.

[61] V.K. Goyal, J. Kovacevic, and M.Vetterli. Quantized frame expansions as source-channel codes for erasure channels. *Proc. IEEE Data Compression Conf., pp. 326-335*, March 1999.

[62] V.K. Goyal and N.T. Thao M. Vetterli. Quantized overcomplete expansions in r^N: Analysis, synthesis and algorithms. *IEEE Trans. Inform. Theory, vol. 44, pp. 16-31*, January 1998.

[63] R.M. Gray. Source coding theory. *Kluwer*, 1990.

[64] R.M. Gray and D.L. Neuhoff. Quantization. *IEEE Trans. Inform. Theory, vol. 44, pp. 2325-2383*, October 1998.

[65] R.M. Gray and A.D. Wyner. Source coding for a simple network. *Bell Syst. Tech. J., vol. 53, no. 9, pp. 1681-1721*, November 1974.

[66] ITU-T Recommendation H.263. *Video coding for low bitrate communication*, 1998.

[67] R. Han, P. Bhagwat, R. LaMaire, T. Mummert, V. Perret, and J. Rubas. Dynamic adaptation in an image transcoding proxy for mobile web browsing. *IEEE Pers. Commun., vol. 5, pp. 8-17*, December 1998.

[68] P. Haskell and D. Messerschmitt. Resynchronization of motion compensated video affected by ATM cell loss. *ICASSP 1992, vol. 3*, March 1992.

[69] Z. He and S.K. Mitra. A unified rate-distortion analysis framework for transform coding. *IEEE Transactions on Circuits and Systems for Video Technology, Vol. 11, No. 12*, December 2001.

[70] R. O. Hinds, T. N. Pappas, and J. S. Lim. Joint block-based video source/channel coding for packet switched networks. *VCIP 1998, SPIE*, January 1998.

[71] J. Hong. Discrete fourier, hartley and cosine transforms in signal processing. *Ph.D. Thesis, Columbia University*, 1993.

[72] S.H. Hong and S.D. Kim. Histogram-based rate-distortion estimation for mpeg-2 video. *IEEE Proc.-Vis. Image Signal Process, Vol. 146, No. 4*, August 1999.

[73] C. Horne and A. R. Reibman. Adaptation to cell loss in a 2-layer video codec for ATM networks. *Picture Coding Symp.*, March 1993.

[74] C. Hsu, A. Ortega, and M. Khansari. Rate control for robust video transmission over wireless channels. *VCIP 1997, SPIE*, February 1997.

[75] K. Illgner and F. Mueller. Spatially scalable video compression employing resolution pyramids. *IEEE Journal on Selected Areas in Communications, Vol. 15, No.9*, December 1997.

[76] A. Ingle and V.A. Vaishampayan. DPCM system design for diversity system with applications to packetized speech. *IEEE Trans. Speech Audio Processing, vol. 3, pp. 48-57*, Januar 1995.

[77] H. Jafarkhani and V. Tarokh. Multiple description trellis-coded quantization. *IEEE Trans. Commun., vol. 47, vol. 47, pp. 799-803*, June 1999.

[78] N.S. Jayant. Subsampling of a dpcm speech channel to provide two 'self-contained' half-rate channels. *Bell Syst. Tech. J., vol. 60, no. 4, pp. 501-509*, April 1981.

[79] N.S. Jayant, E. Y. Chen, J. D. Johnston, S.R. Quackenbush, S.M. Dorward, K. Thompson, R.L. Cupo, J.-D. Wang, C.-E.W. Sundberg, and N. Seshadri. The AT&T in-band adjacent channel system for digital audio broadcasting. *Proc. Int. Symp. Digital Audio Broadcasting*, March 1994.

[80] N.S. Jayant and S.W. Christensen. Effects of packet losses in waveform coded speech and improvements due to an odd-even sample-interpolation procedure. *IEEE Trans. Commun., vol. 29, pp. 101-109*, Februar 1981.

[81] J.Barros and S.D. Servetto. Sequencing multiple descriptions. *IEEE Data Compression Conference, DCC 2002*, 2002.

[82] W. Jiang and A. Ortega. Multiple description coding via ployphase transform and selective quantization. *Proc. SPIE Conf. Visual Commun. and Image Processing, vol. 3653, pp. 998-1008*, 1999.

[83] W. Jiang and A. Ortega. Multiple description speech coding for robust communication over lossy packet networks. *IEEE Int. Conf. Multimedia & Expo, vol. 1, pp. 444-447*, July 2000.

[84] J.A. Kelner, V.K. Goyal, and J. Kovacevic. Multiple description lattice vector quantization: Variations and extensions. *Proc. IEEE Data Compression Conf., pp. 480-489*, March 2000.

[85] V. Koshelev. Multilevel source coding and data transmission theorem. *Proc. VII All-Union Conf. Theory of Coding and Data Transmission, pt. 1, pp. 85-92*, 1978.

[86] V. Koshelev. Hierarchical coding of discrete sources. *Probl. Peredachi. Inf., vol. 16, no. 3, pp. 31-49*, 1980.

[87] V. Koshelev. An evaluation of the average distortion for discrete scheme of sequential approximation. *Probl. Peredachi. Inf., vol. 17, no. 3, pp. 20-33*, 1981.

[88] G. Kubin and W.B. Kleijn. Multiple description coding (MDC) of speech with an invertible auditory model. *Proc. IEEE Workshop Speech Coding, pp. 81-83*, June 1999.

[89] J. Lee and B.W. Dickinson. Temporally adaptive motion interpolation exploiting temporal masking in visual perception. *IEEE Transactions on Image Processing, Vol. 3, No. 5*, September 1994.

[90] K.-W. Lee, R. Puri, T. Kim, K. Ramchandran, and V. Bharghavan. An integrated source coding and congestion control framework for video streaming in the internet. *Proc. IEEE Infocom, vol. 2, pp. 747-756*, March 2000.

[91] W.S. Lee, M.R. Pickering, M.R. Frater, and J.F.Arnold. A robust codec for transmission of very low bitrate video over channels with bursty errors. *IEEE Trans. Circuits Syst. Video Technol., vol 10, pp. 1403-1412*, December 2000.

[92] J.Y. Liao and J.D. Villasenor. Adaptive intra update for video coding over noisy channels. *ICIP 1996*, March 1992.

[93] T. Linder, R. Zamir, and K. Zeger. The multiple description rate region for high resolution source coding. *Proc. IEEE Data Compression Conf. and pp. 149-158*, March 1998.

[94] A. Luthra and P. Topiwala. Overview of the H.264/AVC video coding standard. *Proceedings of the SPIE, vol. 5203*, 2004.

[95] A. Majumdar, R. Puri, K. Ramchandran, and I. Kozintsev. Robust video multicast under rate and channel variability with applications to wireless lans. *IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2002.

[96] S. McCanne, V. Jacobson, and M. Vetterli. Receiver-driven layered multicast. *Proc. ACM SIGCOMM, pp. 117-130*, August 1996.

[97] S. Mehrotra and P.A. Chou. On optimal frame expansions for multiple description quantization. *Proc. IEEE Int. Symp. Information Theory, p. 176*, June 2000.

[98] A.C. Miguel, A.E. Mohr, and E.A. Riskin. SPIHT for generalized multiple description coding. *Proc. IEEE Int. Conf. Image Processing, vol. 3, pp. 842-846*, October 1999.

[99] S.E. Miller. Fail-safe transmission without standby facilities. *Bell Labs, Tech. Rep. TM80-136-2*, August 1980.

[100] S.E. Miller. New transmission configuration. *Bell Labs, Lab Notebook #55637*, May 1978.

[101] A. E. Mohr, E.A. Riskin, and R.E. Ladner. Unequal loss protection: Graceful degradation of image quality over packet erasure channels through forward error correction. *IEEE J. Select. Areas Commun., vol. 18, pp. 819-828*, June 2000.

[102] A. E. Mohr, E.A. Riskin, and R.E. Ladner. Generalized multiple description coding through unequal loss protection. *Proc. IEEE Int. Conf. Image Processing, vol.1, pp. 411-415*, October 1999.

[103] M.T. Orchard, Y. Wang, V. Vaishampayan, and A.R. Reibman. Redundancy rate-distortion analysis of multiple description coding using pairwise correlating transforms. *Proc. IEEE Int. Conf. Image Processing, vol. I, pp. 608-611*, October 1997.

[104] A. Ortega and K. Ramchandran. Rate-distortion methods for image and video compression. *IEEE Signal Processing Mag., vol 15, pp. 23-50*, November 1998.

[105] L. Ozarow. On a source-coding problem with two channels and three receivers. *Bell Syst. Tech. J., vol. 59, no. 10, pp. 1909-1921*, December 1980.

[106] H.C. Papadopoulous and C.-E.W. Sundberg. Simultaneous broadcasting of analog fm and digital audio signals by means of adaptive precanceling techniques. *IEEE Trans. Commun.,vol. 46, pp. 1233-1242*, September 1998.

[107] A. Papoulis. Probability, random variables and stochastic processes, $3^{rd}ed$. *McGraw-Hill*, 1991.

[108] V. Paxson. End-to-end internet packet dynamics. *Proc. of the ACM SIGCOMM, pp. 139-152*, September 1997.

[109] A. Puri, R. Aravind, B.G. Haskell, and R. Leonardi. Video coding with motion-compensated interpolation for CD-ROM applications. *Signal Processing: Image Communication 2 (1990)*, 1990.

[110] R. Puri and K. Ramchandran. Multiple description source coding using forward error correction. *Conf. Rec. 33rd Asilomar Conf. Sig. Sys. Computers, vol. 1, pp. 342-346*, October 1999.

[111] Q.F.Zhu and L. Kerofsky. Joint source coding, transport processing and error concealment for H.323-based packet video. *VCIP 1999, SPIE*, January 1999.

[112] M. Quirk. Diversity coding for communication systems. *Bell Labs, Engineer's Notes (not archived)*, December 1979.

[113] K. Ramchandran, A. Ortega, M. Uz, and M. Vetterli. Multiresolution broadcast for digital HDTV using joint source/channel coding. *IEEE J. Select. Areas Commun., vol 11, pp. 6-23*, Januar 1993.

[114] S. Rangan and V.K. Goyal. Recursive consistent estimation with bounded noise. *IEEE Trans. Inform. Theory, vol. 47, pp. 457-464*, Januar 2001.

[115] E. Reed and J.S. Lim. Optimal multidimensional bit-rate control for video communication. *IEEE Transactions on Image Processing, Vol. 11, No. 8*, August 2002.

[116] A. R. Reibman, H. Jafarkhani, Y. Wang, M.T. Orchard, and R. Puri. Multiple description coding for video using motion compensated prediction. *Proc. IEEE Int. Conf. Image Processing, vol. 3, pp. 837-841*, October 1999.

[117] D.O. Reudink. The channel splitting problem with interpolative coders. *Bell Labs, Tech. Rep. TM80-134-1*, October 1980.

[118] B. Rimoldi. Successive refinement of information: Characterization of the achievable rates. *IEEE Trans. Inform. Theory., vol. 40, pp. 253-259*, January 1994.

[119] D.G. Sachs, A. Raghavan, and K. Ramchandran. Wireless image transmission using multiple-description based concatenated codes. *Proc. SPIE Image Video Processing, vol. 3974, pp. 300-311*, January 2000.

[120] A. Said and W.A. Pearlman. A new fast and efficient codec based on set partioning in hierarchical trees. *IEEE Trans. Circuits Syst. Video Technol., vol. 6, pp. 243-250*, June 1996.

[121] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. Rtp: A transport protocol for real-time applications. *RFC 1989*, January 1996.

[122] M. Schwartz. Information transmission, modulation and noise. *McGraw-Hill,* $4^{th}ed.$, 1990.

[123] S.D. Servetto and K. Nahrstedt. Broadcast-quality video over IP. *IEEE Trans. Multimedia, vol.3, pp. 162-173*, March 2001.

[124] S.D. Servetto and K. Nahrstedt. Video streaming over the public internet: Multiple description codes and adaptive transport protocols. *Proc. IEEE Int. Conf. Image Processing, vol. 3, pp. 85-89*, October 1999.

[125] S.D. Servetto, K. Ramchandran, V. Vaishampayan, and K. Nahrstedt. Multiple description wavelet based image coding. *IEEE Trans. Image Processing, vol.9, pp. 813-826*, May 2001.

[126] S.D. Servetto, V. Vaishampayan, and N.J.A. Sloane. Multiple description lattice vector quantization. *Proc. IEEE Data Compression Conf., pp. 13-22*, March 1999.

[127] C.E. Shannon. Coding theorems for a discrete source with a fidelity criterion. *IRE Int. Conv. Rec., part 4, vol. 7, pp. 93-126*, 1960.

[128] R. Singh, A. Ortega, L. Perret, and W. Jiang. Comparison of multiple description coding and layered coding based on network simulations. *Proc. SPIE Image Video Proc., pp. 929-939*, January 2000.

[129] A. Skodras, C. Christopoulos, and T. Ebrahimi. The JPEG2000 still image compression standard. *IEEE Signal Processing Mag., vol. 18, pp. 36-58*, September 2001.

[130] M. Srinivasan. Iterative decoding of multiple descriptions. *Proc. IEEE Data Compression Conf., pp. 463-472*, March 1999.

[131] E. Steinbach, N. Faerber, and B. Girod. Standard compatible extension of H.263 for robust video transmission in mobile environments. *IEEE Trans. Circuits Syst. Video Technol.*, December 1997.

[132] T. Stockhammer, D. Kontopodis, and T. Wiegand. Rate-distortion optimization for JVT/H.26L video coding in packet loss environment. *12th International Packet Video Workshop (PV 2002)*, May 2002.

[133] K. Stuhlmueller, N. Faerber, M. Link, and B. Girod. Analysis of video transmission over lossy channels. *IEEE Journal on Selected Areas in Communications, Special Issue on Error Resilient Image and Video Transmission, pp. 1012-1032, 18(6)*, June 2000.

[134] P. Subrahmanya and T. Berger. Multiple description encoding of images. *Proc. IEEE Data Compression Conf., p. 470*, March 1997.

[135] G. J. Sullivan and T. Wiegand. Rate-distortion optimization for video compression. *IEEE Signal Processing Mag., vol 15, pp. 74-90*, November 1998.

[136] H. Sun and W. Kwok. Concealment of damaged block transform coded images using projections onto convex sets. *IEEE Transactions on Image Processing, vol. 4, pp. 470-477*, April 1995.

[137] R. Swann and N. G. Kingsbury. The erec: An error resilient technique for coding variable-length blocks of data. *IEEE Trans. Image Processing, vol. 5, pp. 656-574*, April 1996.

[138] W. Tan and A. Zakhor. Real-time internet video using error resilient scalable compression and TCP-friendly transport protocol. *IEEE Transactions on Multimedia, pp. 172-186*, June 1999.

[139] W. Tan and A. Zakhor. Error control for video multicast using hierarchical FEC. *ICIP, pp. 401-405*, October 1999.

[140] T. Turletti and C. Huitema. Videoconferencing on the internet. *IEEE/ACM Trans. Networking, vol. 4, no.3, pp. 340-351*, June 1996.

[141] B.E. Usevitch. A tutorial on modern lossy wavelet image compression: Foundations of JPEG2000. *IEEE Signal Processing Mag., vol. 18, pp. 22-35*, September 2001.

[142] V. Vaishampayan. Design of multiple description scalar quantizers. *IEEE Trans. Inform. Theory, vol. 39, pp. 821-834*, May 1993.

[143] V. Vaishampayan and J.-C. Batllo. Asymtotic analysis of multiple description quantizers. *IEEE Trans. Inform. Theory, vol. 44, pp. 278-284*, January 1998.

[144] V. Vaishampayan and J. Domaszewicz. Design of entropy-constrained multiple description scalar quantizers. *IEEE Trans. Inform. Theory, vol. 40, pp. 245-250*, January 1994.

[145] V. Vaishampayan and S. John. Balanced interframe multiple description video compression. *Proc. IEEE Inf. Conf. Image Processing, pp. 812-816*, October 1999.

[146] V. Vaishampayan and S. John. Interframe balanced-multiple-description video compression. *ICIP*, October 1999.

[147] V. Vaishampayan and A.A. Siddiqui. Speech predictor design for diversity communication systems. *Proc. IEE Workshop Speech Coding for Telecommunications, pp. 69-70*, September 1995.

[148] V. Vaishampayan, N.J.A. Sloane, and S.D. Servetto. Multiple description vector quantization with lattice codebooks: Design and analysis. *IEEE Trans. Inform. Theory, 47(5), pp. 1718-1734*, 2001.

[149] N. Varnica, M. Fleming, and M. Effros. Multi-resolution adaptation of the SPIHT algorithm for multiple description. *Proc. IEEE Data Compression Conf., pp. 303-312*, March 2000.

[150] R. Venkataramani, G. Kramer, and V.K. Goyal. Multiple description coding with many channels. *IEEE Trans. Inform. Theory, vol. 49, no.9, pp. 2106-2114*, September 2003.

[151] M. Vetterli. Wavelets, approximation and compression. *IEEE Signal Processing Mag., vol. 18, pp. 59-73*, September 2001.

[152] J. Walrand. Communication networks: A first course. *MA:McGraw-Hill, 2nd ed.*, 1998.

[153] Y. Wang and S. Lin. Error-resilient video coding using multiple description motion compensation. *IEEE Transactions on Circuits and Systems for Video Technology, Vol. 12, No. 6*, June 2002.

[154] Y. Wang, M.T. Orchard, and A.R.Reibman. Multiple description image coding for noisy channels by pairing transform coefficients. *Proc.IEEE Workshop on Multimedia Signal Processing, pp. 419-424*, June 1997.

[155] Y. Wang, M.T. Orchard, V. Vaishampayan, and A.R. Reibman. Multiple description coding using pairwise correlating transforms. *IEEE Trans. Image Processing, vol. 10, pp. 351-366*, March 2001.

[156] Y. Wang and Q. Zhu. Error control and concealment for video communications: A review. *Proceedings of the IEEE, pp. 974-997*, May 1998.

[157] Y. Wang and Q. F. Zhu. Maximally smooth error recovery in transform coding. *IEEE Trans. Commun., vol. 41, pp. 1544-1551*, October 1993.

[158] S. Wenger and G. Cote. Using RFC2429 and H.263+ at low to medium bitrates for low-latency applications. *Packet Video Workshop 1999*, April 1999.

[159] S. Wenger, G. Knorr, J. Ott, and F. Kossentini. Error resilience support in H.263+. *IEEE Trans. Circuits Syst. Video Technol., vol. 8, pp. 867-877*, November 1998.

[160] T. Wiegand, M. Lightstone, D. Mukherjee, T. George, and S. K. Mitra. Rate-distortion optimized mode selection for very low bit rate video coding and the emerging H. 263 standard. *IEEE Trans. Circuits Syst. Video Technol., vol. 6, pp. 182-190*, April 1996.

[161] H.S. Witsenhausen. On source networks with minimal breakdown degradation. *Bell Syst. Tech. J., vol. 59, no. 6, pp. 1083-1087*, August 1980.

[162] H.S. Witsenhausen. An achievable region for the breakdown degradation problem with multiple channels. *Bell Labs, Tech. Rep. TM81-11217-3*, January 1981.

[163] H.S. Witsenhausen and A.D. Wyner. Source coding for multiple descriptions II: A binary source. *Bell Labs, Tech. J., vol. 60, no. 10, pp. 2281-2292*, December 1981.

[164] J.K. Wolf, A.D. Wyner, and J. Ziv. Source coding for multiple descriptions. *Bell Syst. Tech. J., vol. 59, no. 8, pp. 1417-1426*, October 1980.

[165] S.-M. Yang and V.A. Vaishampayan. Low-delay communication for rayleigh fading channels: An application of the multiple description quantizer. *IEEE Trans. Commun., vol. 43, pp. 2771-2783*, November 1995.

[166] X. Yang and K. Ramchandran. Optimal subband filter banks for multiple description coding. *IEEE Trans. Inform. Theory, vol. 46, pp. 2477-2490*, November 2000.

[167] R. Zamir. Gaussian codes and shannon bounds for multiple descriptions. *IEEE Trans. Inform. Theory, vol. 45, pp. 2629-2635*, November 1999.

[168] A. Zandi, J.D. Allen, E.L. Schwartz, and M. Boliek. CREW: Compression with reversible embedded wavelets. *Proc. IEEE Data Compression Conf., pp. 212-221*, March 1995.

[169] R. Zhang, S. L. Regunathan, and K. Rose. Video coding with optimal inter/intra-mode switching for packet loss resilience. *IEEE Journal on Selected Areas in Communications, Vol. 18, No. 6*, June 2000.

[170] R. Zhang, S. L. Regunathan, and K. Rose. Optimal intra/inter mode switching for robust video communication over the internet. *33rd Asilomar Conf. Signals, Syst., Computers*, October 1999.

[171] Z. Zhang and T. Berger. New results in binary multiple descriptions. *IEEE Trans. Inform. Theory, vol. 33, no. 4, pp. 502-521*, July 1987.