Evaluation of the H.264 codec

Internal Report

Carlos Miguel Tavares Calafate, Manuel Pérez Malumbres

DISCA - UPV - 2003

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Digital video is being adopted in an increasingly proliferating array of applications ranging from video telephony and videoconferencing to DVD and digital TV. The adoption of digital video in many applications has been fueled by the development of efficient video coding standards, resulting in many video coding standards that cover a wide range of application areas. These standards were defined taking into account the interoperability between systems designed by different manufacturers for any given application, hence facilitating the growth of the video market. The ITU-T[1] is now one of two most important standard organizations that develop video coding standards - the other being ISO/IEC JTC1[2]. The ITU-T video coding standards are called recommendations, and they are denoted with H.26x (e.g., H.261, H.262, H.263 and H.264). The ISO/IEC standards are denoted with MPEG-x (e.g., MPEG-1, MPEG-2 and MPEG-4). Most ITU-T recommendations have been designed for real-time video communication applications, such as video conferencing and video telephony. On the other hand, the MPEG standards have been designed to address the requirements of video storage (DVD), broadcast video (broadcast TV), and video streaming (e.g., video over the Internet, video over DSL, video over wireless) applications. Both standardization bodies have worked independently on the different standards. The only exception has been the H.262/MPEG-2 standard, which was developed jointly by the two committees. Recently, the ITU-T VCEG and the ISO/IEC JTC1 have agreed to join their efforts in the development of the emerging H.264 standard, which was initiated by the ITU-T committee as H.26L. H.264 is being adopted by both committees because they were working with very similar techniques for developing extensions of current standards in order to significantly increase coding performance. So, joining efforts to exchange ideas and develop a common framework was the main reason for working together. Figure 1.1 summarizes the evolution of the ITU-T recommendations and the ISO/IEC MPEG standards. Please see [3] and [4] for more information on the H.263 and MPEG-4 video coding standards.

The purpose of this work is to evaluate the performance of this new standard in error-prone environments where packet losses often occur. The strategy we followed was to start from a simple configuration where all the options were disabled by default, and from there evaluate the performance of each of the parameters of interest. In a second stage the encoder was tuned using the results achieved in the previous stage, picking the best values in terms of rate-distortion performance. During this second stage, parameters specific to error resilience were tested by simulating random packet losses.

Our work proceeded with simulations of packet loss bursts, in order to test the tools that focus on this problem. These tools are flexible macroblock ordering for short bursts and multi-frame prediction for

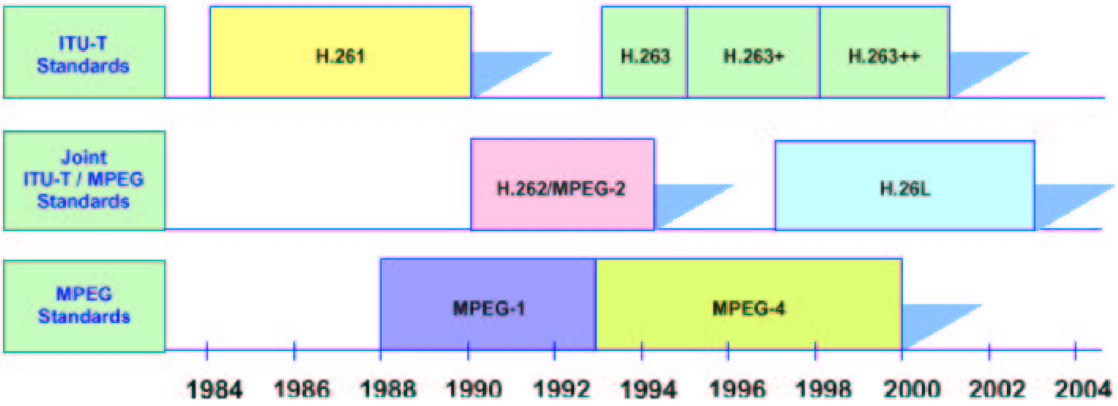


Figure 1.1: Progress of ITU-T recommendations and MPEG standards (from MPEG group web site[5]).

longer bursts.

Chapter 2

H.264 overview

The main objective behind the H.264 project is to develop a high-performance video coding standard by adopting a back to basics approach where simple and straightforward design using well-known building blocks is used. The ITU-T Video Coding Experts Group (VCEG) has initiated the work on the H.26L standard in 1997. Towards the end of 2001, and witnessing the superiority of video quality offered by H.26L-based software over that achieved by the existing most optimized MPEG-4 based proposals, ISO/IEC MPEG joined ITU-T VCEG by forming a Joint Video Team (JVT) that took over the H.26L project of the ITU-T. The JVT objective is to create a single video coding standard that will simultaneously result in a new part (Part-10) of the MPEG-4 family of standards and a new ITU-T (H.264) Recommendation. The H.264 development work is an on-going activity, with the first version of the standard expected to be finalized technically before 2002 and officially before the end of the year 2003. The emerging H.264 standard has a number of features that distinguish it from existing standards, while at the same time, sharing common features with them.

Some of the key features of H.264 are: up to 50% in bit rate savings, high quality video (including low bit rates), adaptation to delay constraints (real-time communications applications as well as video storage, sever-based video streaming applications), error resilience tools to deal with packet loss in packet networks, and bit errors in error-prone wireless networks. Network friendliness is also aimed by this standard through a conceptual separation between a Video Coding Layer (VCL), which provides the core high-compression representation of the video picture content, and a Network Adaptation Layer (NAL), which packages that representation for delivery over a particular type of network.

The above features can be translated into a number of advantages for different video applications.

The underlying approach of H.264 is similar to that adopted in previous standards such as H.263 and MPEG-4, and consists of the following four main stages:

1. Dividing each video frame into blocks of pixels, so that processing of the video frame can be conducted at the block level. by using the well known DCT transform.
2. Exploiting the spatial redundancies that exist within the video frame by coding some of the original blocks through transform, quantization and entropy coding (or variable-length coding).
3. Exploiting the temporal dependencies that exist between blocks in successive frames, so that only changes between successive frames need to be encoded. This is accomplished by using motion estimation and compensation. For any given block, a search is performed in the previously coded

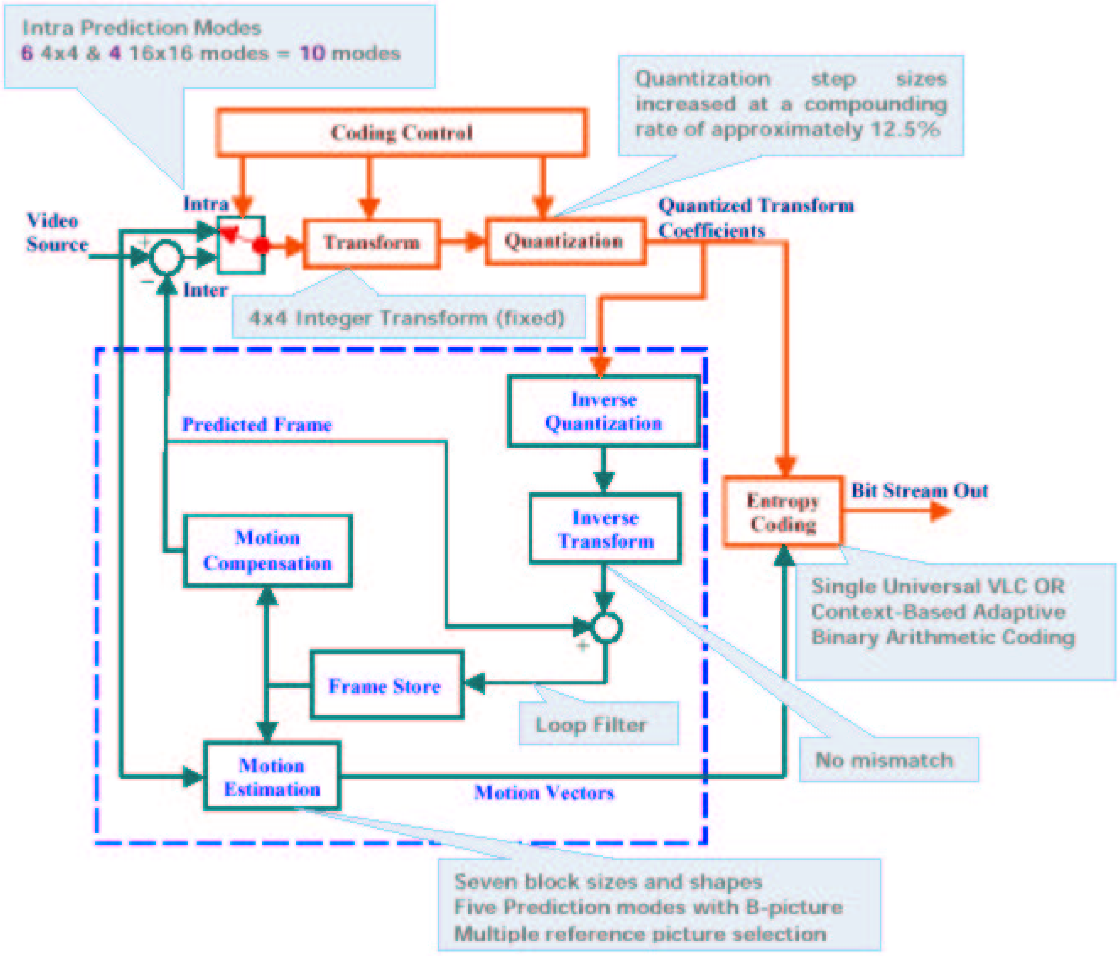


Figure 2.1: Block diagram of the H.264 encoder

one or more frames to determine the motion vectors that are then used by the encoder and the decoder to predict the subject block.

4. Exploiting any remaining spatial redundancies that exist within the video frame by coding the residual blocks, i.e., the difference between the original blocks and the corresponding predicted blocks, again through transform, quantization and entropy coding.

From the coding point of view, the main differences between H.264 and the other standards are summarized in figure 2.1 through an encoder block diagram. From the motion estimation/compensation side, H.264 employs blocks of different sizes and shapes, higher resolution sub-pixel motion estimation, and multiple reference frame selection. In the transform side, H.264 uses an integer based transform that approximates the DCT transform used in previous standards, but does not have the mismatch problem in the inverse transform. In H.264, entropy coding can be performed using either a single Universal Variable Length Codes (UVLC) table or using Context-based Adaptive Binary Arithmetic Coding (CABAC).

Chapter 3

Codec evaluation and tuning in a lossless environment

In this chapter we evaluate the parameters of interest related to the H.264 framework. The only restrictions were set by the selected version of the reference software, which is JM2.0. These restrictions, however, were essentially related with robustness under packet loss (ex.: FMO reordering), so they will be exposed in later chapters.

Concerning the test sequences used, these were the well known News and Foreman sequences. The need for different test sequences is related to different levels of movement between both, which produce at times different conclusions.

3.1 Intra Frame Period

Intra frames are coded independently from other frames, using simply what is called intra coding. This coding is similar to what is used for static pictures which provides methods which simply take care of spatial redundancy issues, and therefore does achieve great compression rates since it does not take advantage of temporal redundancy. They reach particular relevance in terms of error propagation, since the introduction of an I frame will stop the propagation of errors, so the prediction process can start again from an error-free situation.

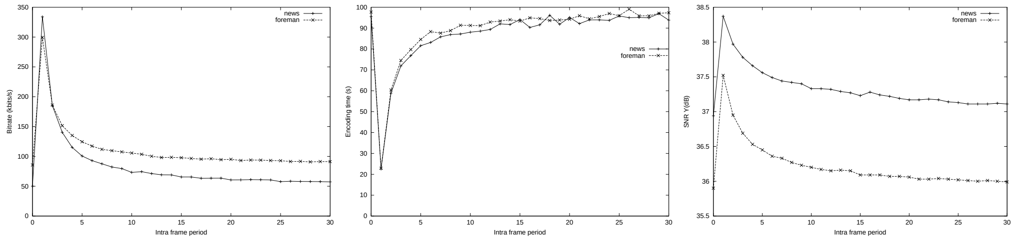


Figure 3.1: Bit rate, encoding time and PSNR variation varying the number of intra pictures.

As can be appreciated in figure 3.1, the introduction of a great number of intra coded frames results in small encoding times but very high bit rates as expected. The peak variations in PSNR are not very significative, and are due mostly to the slight differences in the quantization parameters for I and P frames.

That figure points out that using a period of 15 (MPEG2-4 GOP size) or more between intra frames has almost no effect in terms of bit-rate and PSNR relative to a solution where no intra frames are used. This aspect is particularly interesting in terms of error-resilience.

3.2 Quantization parameter for P frames

The evaluation of P frames was done using only a single I frame at the beginning. Figure 3.2 shows that the encoding time is quite stable and that PSNR decreases linearly with increasing quantization as expected. It is the bit rate curve which gives more relevant data since non-linearities appear. The quantization value of 15 seems to be the frontier between a zone where the actual data is dominant and a zone where other data such as RTP and stream headers dominate the bit-rate curve.

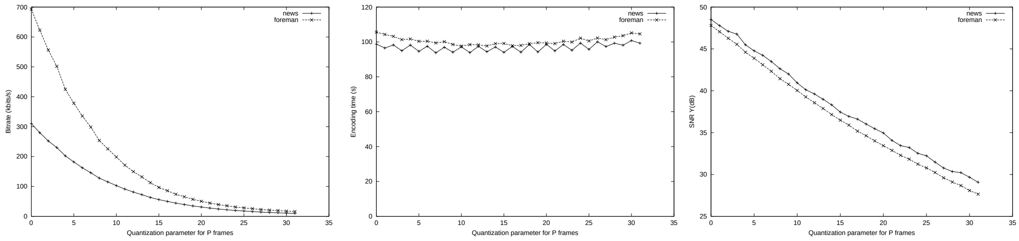


Figure 3.2: Bit rate, encoding time and PSNR variation varying the quantization for P frames

This bit-rate non-linearity relative to P frames will have effects essentially in terms of choosing the best option for streaming. In general, quantization values between 15 and 25 seem appropriate in terms of both bit-rate and PSNR.

3.3 Search Range

The search range sets the searching area in what refers to motion estimation methods. As it can be seen in figure 3.3, the encoding time grows exponentially with increasing search range and PSNR variations are minimal because the quantization values are not changed. We can see that the bit rate is affected, being lower as search range increases. That behavior is due to the required bits for coding motion vectors. As search range decreases more motion vectors will be coded, and as a consequence a larger bit-rate will be produced. However, we can see that for each sequence there is a threshold after which the bit rate seems to remain stable and also this threshold depends on the movement degree present in the sequence. So sequences with little movement, such as the news sequence, need only a very small search area for optimal results.

This fact can be clearly seen in the PSNR plot, where the difference between the minimum and maximum values is about 0.1 dB for the News sequence and about 0.2 dB for the Foreman sequence (higher level of movement). Also, the News sequence reaches a value close to maximum with a search range of 4, while the Foreman with a search range of 7.

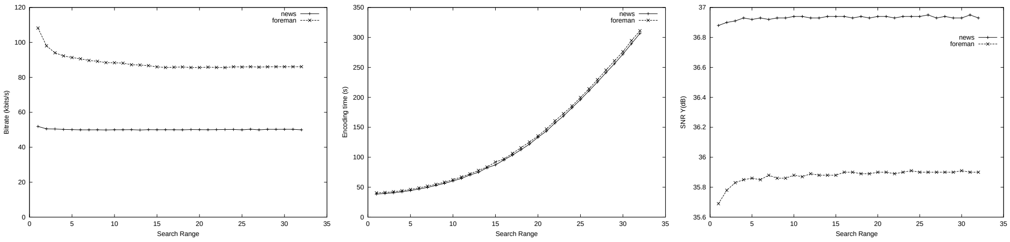


Figure 3.3: Search Range

3.4 Number of Reference Frames

Multiple reference frames are used in the motion estimation process in order to obtain better results in terms of both compression and robustness.

The H.264 standard offers the option of employing multiple reference frames in inter picture coding. Up to five different reference frames can be selected, resulting in better subjective video quality and more efficient coding of the video sequence. Moreover, using multiple reference frames might help making the H.264 bit stream error resilient. As can be seen in figure 3.4, using multiple reference frames significantly increases the complexity since the encoding time will increase proportionally to the number of reference frames used. We have therefore a trade off between bit-rate/PSNR and encoding time which deserves scrutiny since the obtained bit-rate reduction is not very significative and the maximum increase in PSNR is about 0.1 dB only (Foreman sequence).

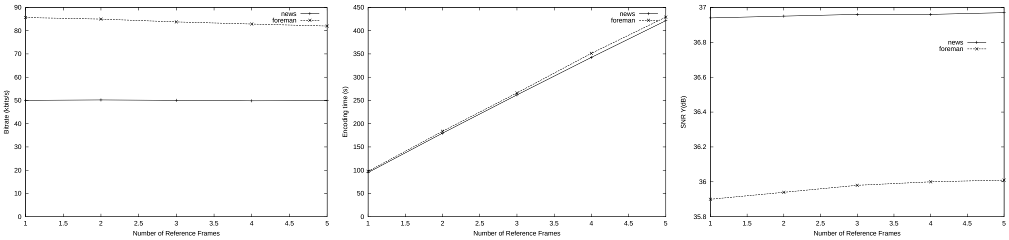


Figure 3.4: Number of reference frames

To support multiple reference frames, both encoder and decoder require much more memory resources. In general, we could say that it is a good choice for asymmetric video distribution (as in DVDs), but its use among devices with limited resources, such as PDAs or mobile phones is prohibitive.

3.5 Macroblock Line Intra Updates

This parameter may be set to Intra coding a GOB (Group of blocks) every N frames. Its usefulness is related to stopping error propagation as in I frames, but now this behavior is distributed throughout the sequence. So, the same effect is achieved at a more constant bit rate.

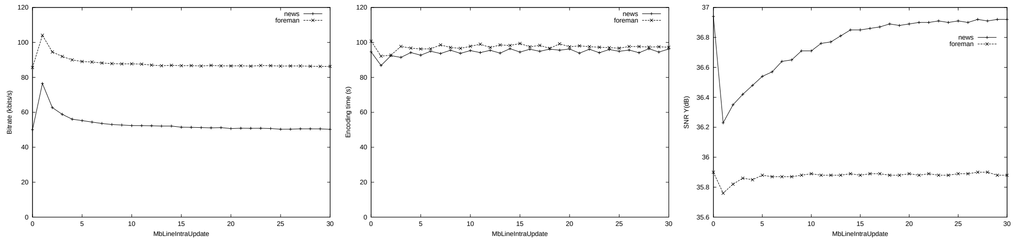


Figure 3.5: Macroblock Line Intra Update

As it can be seen at figure 3.5, after a period of 10 the bit-rate is almost unchanged, though differences of a few tenths of dB can still be noticed for the Foreman sequence. The results relative to PSNR also evidence that the quantization values used in the intra update process are different from those globally defined. Should the PSNR be pushed to a same level, the peak experienced in terms of bit-rate would be even higher.

3.6 B pictures

3.6.1 Quantization parameter for B pictures @ 30 Hz

The use of B pictures for higher frame rates makes sense for high quality video encoding systems. However, encoding B frames significantly increases the encoding time, which requires a high performance processor system. As shown in figure 3.6, the use of B pictures increases significatively the encoding time, which mean that a highly efficient codec will have to be design for the support of B pictures for devices like PDAs or mobile phones.

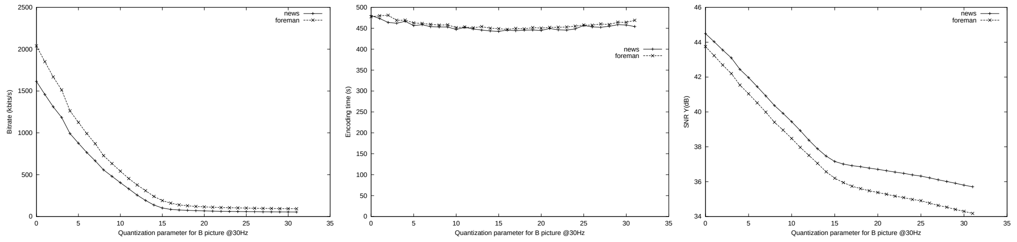


Figure 3.6: Quantization parameter for B pictures

Bit rate and PSNR graphs show that there is a “knee effect” when quantization values reach 15, leading us to consider that for values above 15 the improvement on bit-rate does not compensate for the loss of PSNR.

3.6.2 B pictures @ 10 Hz

The use of B pictures allows the encoder to achieve much greater compression values than regular P pictures do. However, each B picture needs an average of 4 times more processing than P pictures do. Also, due to the multi-frame nature of the prediction performed on B pictures, the transmission of real time video can be affected since each B frame will need to reference future P or I pictures on compression and decompression. This will restrict it's use severely, contrary to what happens on stored video where large GOPs with abounding B pictures are commonly used.

As shown in figure 3.7, mixing B frames will produce substantially better results, even in the simplest case (left).

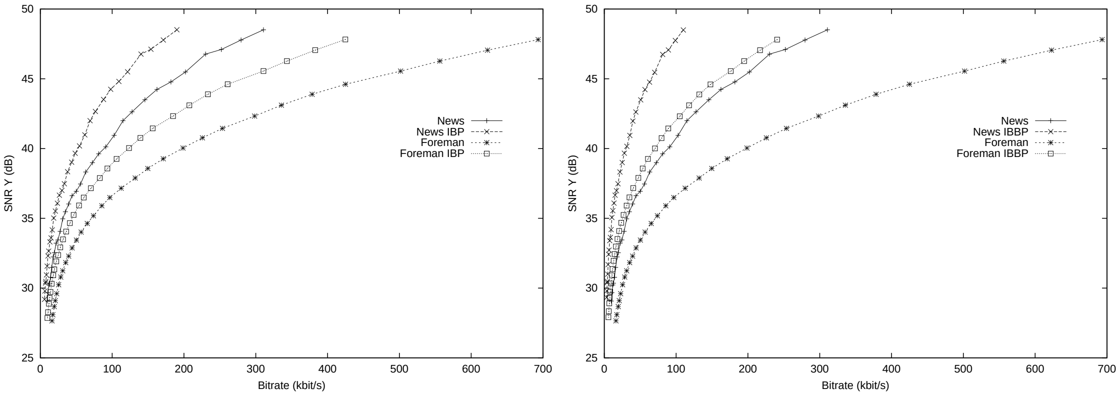


Figure 3.7: Results with the introduction of 1 (left) and 2 (right) B frames on the GOP.

In general the use of B frames for real-time video streaming will be restricted due to the temporal restrictions that have to be met on the encoder. They can be used more conveniently to allow temporal scalability, as presented previously.

3.7 SP Picture Interval

There are two types of S-pictures, namely SP-pictures and SI-pictures. SP-pictures make use of motion compensated predictive coding to exploit temporal redundancy in the sequence similar to P-pictures and SI-pictures make use of spatial prediction similar to I-pictures. Unlike P-pictures, however, SP-picture coding allows identical reconstruction of a frame even when different reference frames are being used. SI picture, on the other hand, can identically reconstruct a corresponding SP-picture. These properties of S pictures provide functionalities for bit-stream switching, splicing, random access, VCR functionalities such as fast-forward and error resilience/recovery.

The H.264 codec currently supports SP pictures only.

As it can be seen in figure 3.8, their effect in terms of bit-rate and PSNR is essentially the same as the one provoked by the introduction of I frames. Therefore, choosing this kind of frames is appropriate in situations where other properties of SP frames, such as random-access, are desired

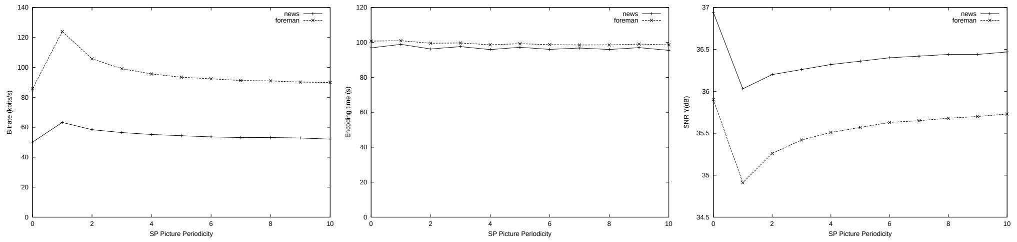


Figure 3.8: SP picture periodicity

3.8 Motion vector resolution

The motion vector resolution sets the granularity allowed in prediction tasks. The H.264 encoder may be tuned to use either 1/4 pixel accuracy or 1/8 of pixel accuracy. As it can be seen in figure 3.9, higher motion vector resolution leads to better results, being more significant the improvement in video sequences with medium or high movement. However, these improvements are only slight and increase the complexity of motion estimation and as a consequence the decoder delay.

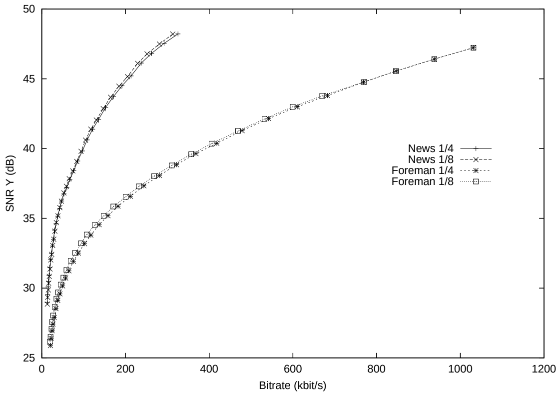


Figure 3.9: 1/4 vs. 1/8 Motion vector resolution

3.9 Hadamard transform

The H.264 encoder allows the use of either the Hadamard transform or the DCT transform, which was the one used by default in previous standards, such as MPEG-2 and H.263. As it can be seen in figure 3.10, the Hadamard transform achieves better results than the DCT, which validates and justifies its integration into the H.264 standard.

This improvement, though, does not surpass 0.5 dB for the same bit rate, which means that perhaps there is little room for improvement using this family of transforms.

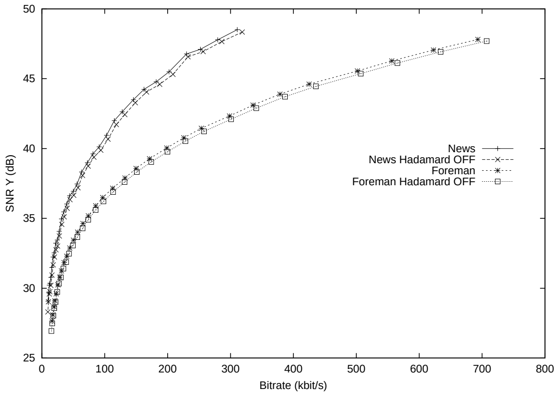


Figure 3.10: Hadamard vs. DCT

3.10 Rate Distortion Optimization

The rate-distortion optimization made available by the H.264 codec consists mainly of an enhancement to the prediction process, so that the codec is able to compare the compression achieved by performing prediction tasks for a certain macroblock, in contrast to the choice of intra coding that same macroblock. As shown in figure 3.11, this method achieves better results for sequences with high levels of movement, even though the gain is small.

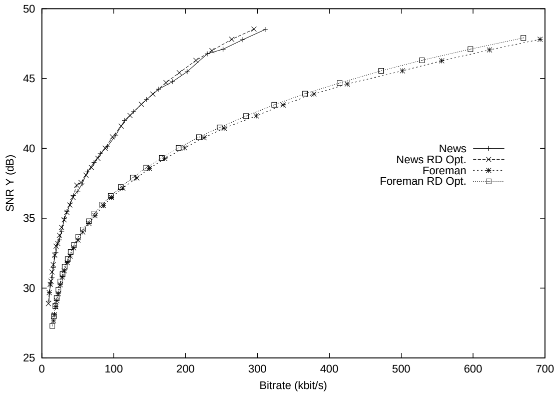


Figure 3.11: Rate Distortion Optimization results

However, in terms of robustness, intra-updated macroblocks are always a good solution. This means that by activating Rate Distortion Optimization there are gains in terms of bit-rate and error-resilience.

3.11 Constrained Intra Prediction

The aim of constraining the intra prediction process is to increase error resilience by avoiding the use of neighboring inter macroblock residual data and decoded samples for the prediction of intra macroblocks. This way the loss of data will not affect negatively intra predicted macroblocks, which results in an effective method to block the propagation of errors. As shown in figure 3.12, the rate-distortion results are only slightly affected by activating this parameter.

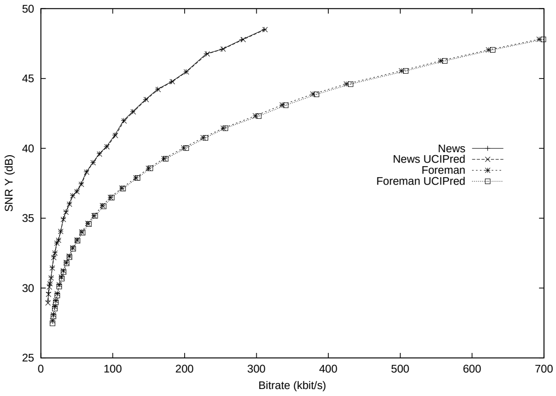


Figure 3.12: Constrained Intra Prediction results

In terms of error-resilience, though, this parameter may show good results, as exposed in the next chapter.

3.12 CABAC vs. UVLC

Universal VLC Entropy coding, based on the use of Variable Length Codes (VLCs), is the most widely used method for the compression of quantized transform coefficients, motion vectors, and other encoder information. VLCs are based on assigning shorter codewords to symbols with higher probabilities of occurrence, and longer codewords to symbols with less frequent occurrences. The symbols and the associated codewords are organized in look-up tables, referred to as VLC tables, which are stored at both the encoder and decoder. In some video coding standards such as H.263, a number of VLC tables are used, depending on the type of data under consideration (e.g., transform coefficients, motion vectors). H.264 offers a single universal VLC table that is to be used in entropy coding of all symbols in the encoder, regardless of the type of data those symbols represent. Although the use of a single UVLC table is simple, is has a major disadvantage, which is that the single table is usually derived using a static probability distribution model, which ignores the correlations between the encoder symbols.

Context-Based Adaptive Binary Arithmetic Coding (CABAC) Arithmetic coding makes use of a probability model at both the encoder and decoder for all the syntax elements (transform coefficients, motion vectors). To increase the coding efficiency of arithmetic coding, the underlying probability model is adapted to the changing statistics within a video frame, through a process called context modeling.

Figure 3.13 shows that using CABAC can result in gains of more than 1 dB for the same bit-rate, or reduce the bit-rate as much as 14,4% for the same distortion (Foreman).

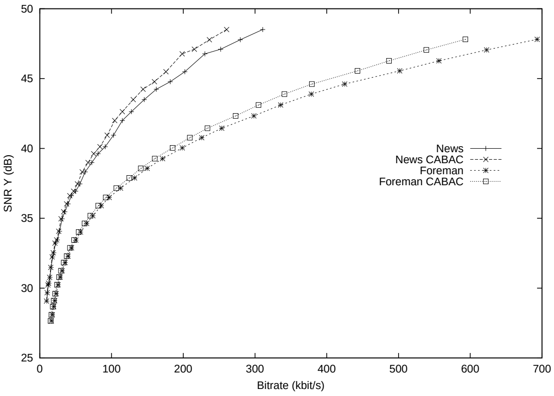


Figure 3.13: CABAC vs. UVLC

Chapter 4

Evaluation of H.264 in random packet loss scenarios

H.264 makes available error resilience mechanisms both on the encoder and on the decoder side. In the encoder we find several parameters that can be tuned. So, a trade-off between compression rate and error resilience can be made targeting different type of problems found in heterogeneous environments.

Random intra macroblock updates and the insertion of intra-coded pictures (I frames) are the most commonly used methods to stop the temporal propagation of errors when no feedback channel is available. While intra frames resets the prediction process, avoiding error propagation, their use has a generally high bandwidth cost causing also severe bit rate variations. The use of random intra macroblock refreshes is more effective than I frames because not only aids to achieve CBR-like streams, but it can also provide better results by statistically reseting the error for each of the macroblocks. The Macroblock Line Intra Update is another robustness option where a group of blocks will be intra coded every N frames. It is just another form of macroblock updating.

The use of slices is another method commonly used in order to improve robustness by stopping spatial error-propagation. The macroblocks belonging to a slice can be decoded independently from other slices since no inter-slice dependencies are allowed. In our work we have used slices intensively since this mechanism is straightly related to the RTP packetization process performed by the encoder.

Another method which deserves consideration is Flexible Macroblock Ordering (FMO), whereby the sender can transmit macroblocks in non-scan order. This method, although similar to slice interleaving, provides much greater flexibility and can be tuned to be more effective in terms of error resiliency. It aims essentially at dealing with packet loss bursts.

SP slices make use of motion-compensated predictive coding to exploit temporal redundancy in the sequences, like P slices do. Unlike P slices, however, SP slice coding allows identical reconstruction of a slice even when different reference pictures are being used. They aim essentially at bit stream switching, splicing, random access, VCR functionalities and error resilience issues.

Another tool that integrates the H.264 codec is Rate Distortion Optimization[6]. Distortion can arise due to either quantization errors and prediction from concealed blocks. If prediction does not provides good compression, intra compression for single Macroblocks is allowed. Concerning encoder tuning, it can be set to OFF for no optimization and ON if such optimization is desired. However, such values will only be optimal in the absence of errors in the network. For that reason, a third mode is available where the encoder takes into account the expected packet loss rate of the network, as well as the decoder's methods to cope with errors in order to decide weather to intra or inter code a block. See [7] for more details on that subject.

The constrained intra prediction option is related with the H.264 intra prediction mode. When it is active avoids using inter macroblock pixels to predict intra macroblocks.

Multi-frame compensation prediction is another tool targeting to increase both compression performance and error resilience, since the loss of an entire reference frame will have less critical effects on later predicted frames [8, 9].

Concerning the decoder, it also plays a fundamental role in error resilience since it is responsible for error concealment tasks. With that purpose it keeps a status map for macroblocks which indicates for each frame being decoded weather a certain macroblock has been correctly received, lost or already concealed. The methods used vary between intra and inter frames. For intra frames the task mainly consists on of performing a weighted pixel averaging on each lost block in order to turn it into a concealed one. For inter frames the task performed consists mainly of guessing the adequate motion vector for lost macroblocks, although intra-style methods can also be used. For a more complete description of such methods please refer to [10].

The decoder also has other tasks like handling multiple reference frames or entire frame losses.

As exposed in [7], the reference decoder for H.264 does not incorporate bit error resilience features since it increases significantly the complexity of the decoder, with only slight improvements as a result. Therefore, bit error detection and handling has to be processed externally.

The evaluation done in this chapter aims verifying the effectiveness of the robustness tools developed under the H.264 framework. To achieve this, we used the version JM3.9a of the H.264 reference software.

In the previous chapter we used the News and Foreman sequences for our evaluation. Now we introduce the Bus sequence which has a higher movement degree than previous test sequences.

4.1 Rate control

In order to perform a detailed evaluation of the H.264 codec, we are going to evaluate a lot of configuration parameters working with different test sequences at different packet loss scenarios.

In figure 4.1 we show the selected performance metrics we are going to employ. Since the H.264 codec does not currently have a rate control mechanism, a program in Perl was created with that aim.

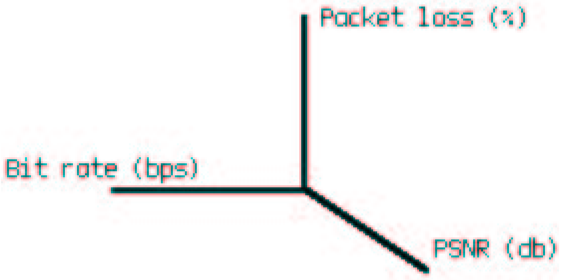


Figure 4.1: Parameters of interest in error-resilience evaluation.

The Perl rate controller is an external mechanism that changes the quantization values until it achieves a value that best matches the bit-rate one entered by the user. The next sections will present not only the error-resilience results, but also the results of the rate controller here described.

4.2 Video Robustness

Our evaluation of the H.264 codec was done using both PSNR measurements and a Robustness parameter (R) that we defined. This parameter, contrarily to PSNR, does not aim at providing a measure of the quality of the sequence, but instead, offers a mean through which the ability to sustain the image quality in the presence of error is quantified.

This parameter is defined by:

$$R = \frac{1}{N} \times \sum_{i=0}^{N} \frac{MSE^{i}_{error-free}}{MSE^{i}} \text{ , } 0 < \text{R} \leq 1$$

In the absence of any kind of error, the Robustness will remain at 1. As the error-rate increases, the R values decrease quadratically.

4.3 Evaluation under loss

4.3.1 Results for I period variation

Figure 4.2 shows the output from the external rate controller, as well as the PSNR results for different intra frame periods. The rate controller's output shows more difficulty to maintain a common bit-rate at higher bit-rate values due to the coarse granularity available. The intensive use of intra-coded frames has a negative impact on the PSNR value in no-loss scenarios, specially for the News sequence that has low levels of movement.

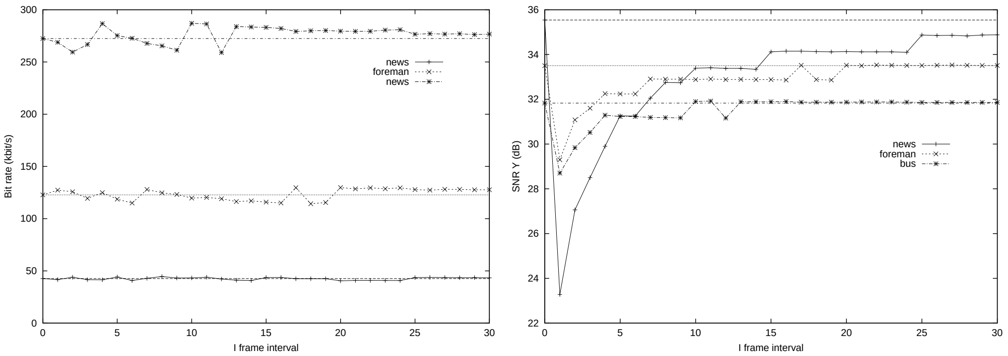


Figure 4.2: Bit rate and PSNR results for loop control with variable period for I frames

However, as shown in figures 4.3 and 4.4, the robustness and distortion results achieved by using intra frames frequently are in general quite superior. As shown, video quality will benefit from the intensive use of intra frames when the packet loss rate is high.

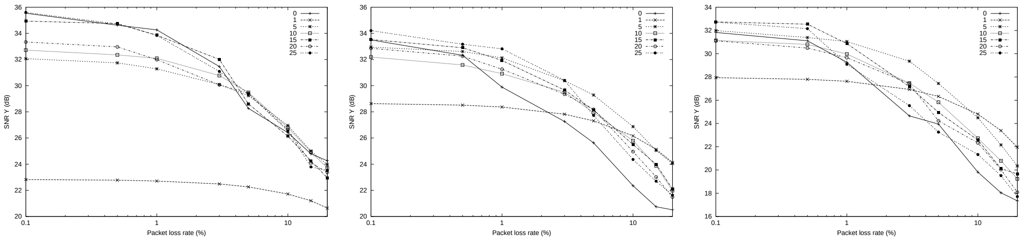


Figure 4.3: PSNR results with different I periods for News (left), Foreman (center) and Bus (right) sequences.

In general there will be an optimal I period for each sequence and for each packet loss rate. A method that would adapt itself to the sequence and network congestion would be optimal. However, in the absence of such method we can say that using an I period ranging from 5 to 15 produces overall good results.

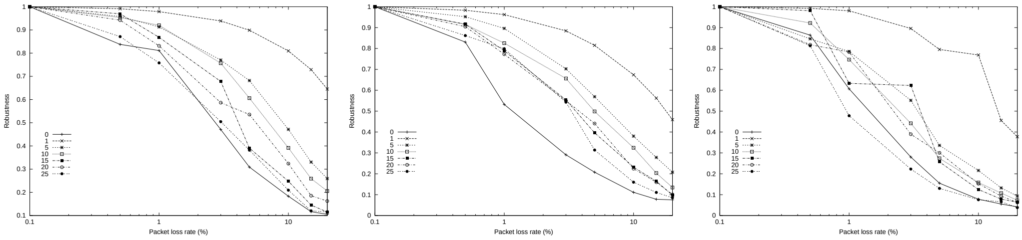


Figure 4.4: Robustness results with different I periods for News (left), Foreman (center) and Bus (right) sequences.

In terms of robustness, there is almost a direct relationship between the intra frames period and the robustness itself as expected.

4.3.2 Rate control results for Constrained Intra Prediction.

As stated in the previous chapter, there is a slight loss in picture quality by constraining the intra prediction process not to use inter pixels. However, concerning error-resilience, it can be a factor that deserves serious consideration since the results achieved on packet-loss scenarios can be much better.

Table 4.1 shows the output of the rate controller and, as was supposed to happen, the use of Constrained Intra Prediction generates worse distortion and higher bit-rate.

Table 4.1: PSNR and rate control results for Constrained Intra Prediction

	News		Foreman		Bus	
	Original	Con CIP	Original	Con CIP	Original	Con CIP
PSNR (dB)	35,54	35,52	33,50	33,46	31,83	31,82
Bit-rate (kbit/s)	42,64	43,26	122,80	122,85	272,46	272,93

As it can be seen by inspecting figures 4.5 and 4.6, there is in general an increased robustness and PSNR over 3-5% of packet losses, which justifies its use in scenarios and networks that suffer from intensive packet losses.

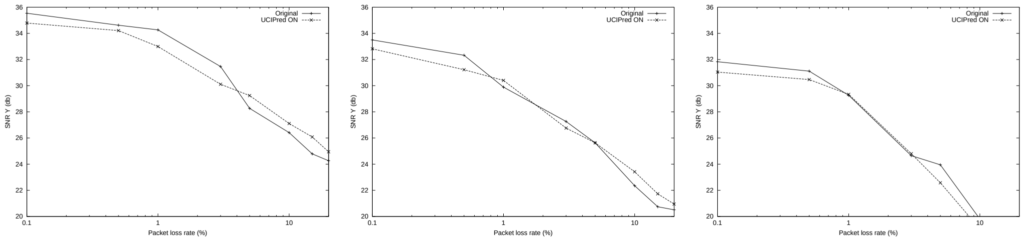


Figure 4.5: PSNR results with packet losses for News (left), Foreman (center) and Bus (right) sequences.

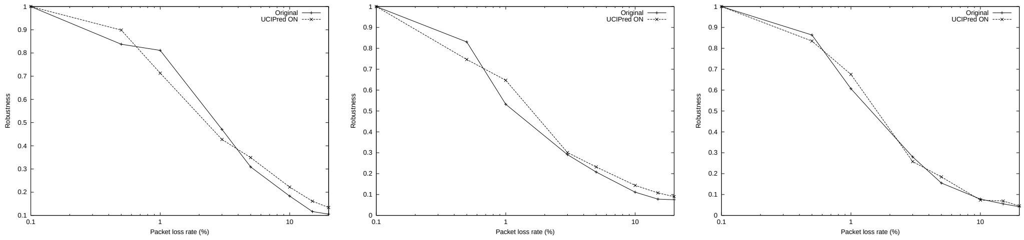


Figure 4.6: Robustness results for News (left), Foreman (center) and Bus (right) sequences.

4.3.3 B frames

Even though the codec does not yet support scenarios with packet losses occurring on B frames, some tests where performed in order to obtain entry-level information on the distortion achieved using bidirectional predicted frames.

As shown in figure 4.7, using more B frames does not always translate into distortion improvements. In particular, for low movement sequences, the use of B frames increases bit-rate but without significative quality improvements. At the other hand, for high movement sequences, as more B frames are used, better quality performance results, because movement will be better encoded. However, there is a bound in the number of B frames (in that case 4) where correlation between both reference frames (I and P) is very low, being B frames useless.

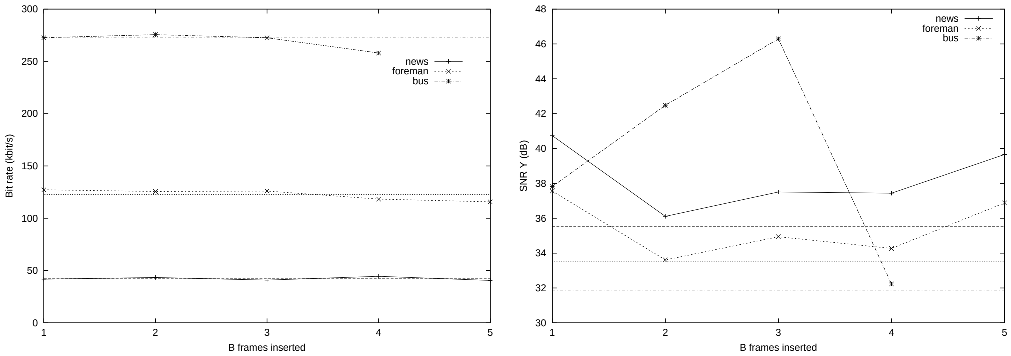


Figure 4.7: Bit rate and PSNR results for loop control with variable number of B frames

Concerning error-resilience, it can be seen that the compression of B frames can lead to better robustness and distortion results, although it will require a per sequence analysis in order to find optimal values. However, as stated previously, these results are not entirely accurate due to a lack of support by the current version of the H.264 reference software concerning losses in B frames. This means that all losses affect only to P frames.

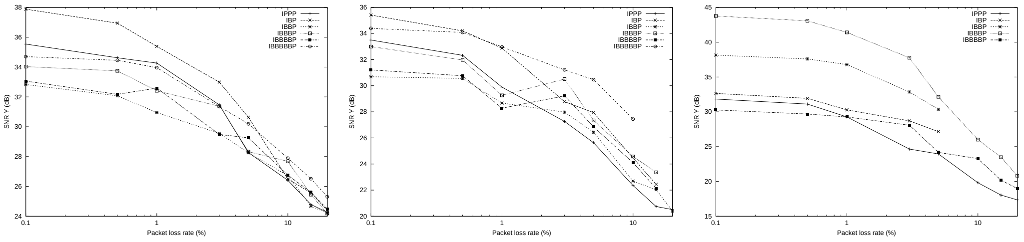


Figure 4.8: PSNR results with packet losses for News (left), Foreman (center) and Bus (right) sequences.

This fact explains why in some of the tests with many B frames we were not able to achieve results under high loss rates.

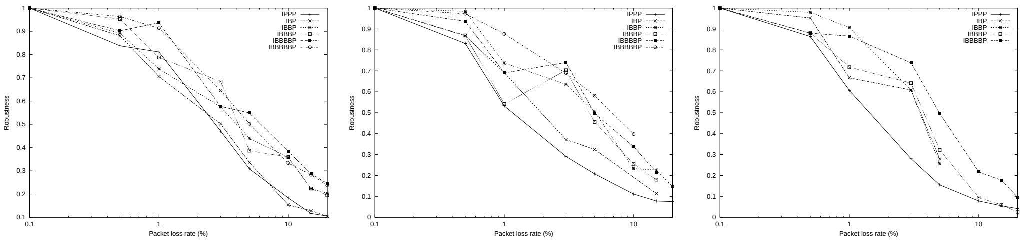


Figure 4.9: Robustness results for News (left), Foreman (center) and Bus (right) sequences.

4.3.4 Macroblock Lines intra update

This parameter offers another method to intra code parts of a frame. Even though no benefits are obtained by using this option in a no-loss environment, the results change when the environment is error prone. The rate controller output is presented in figure 4.10.

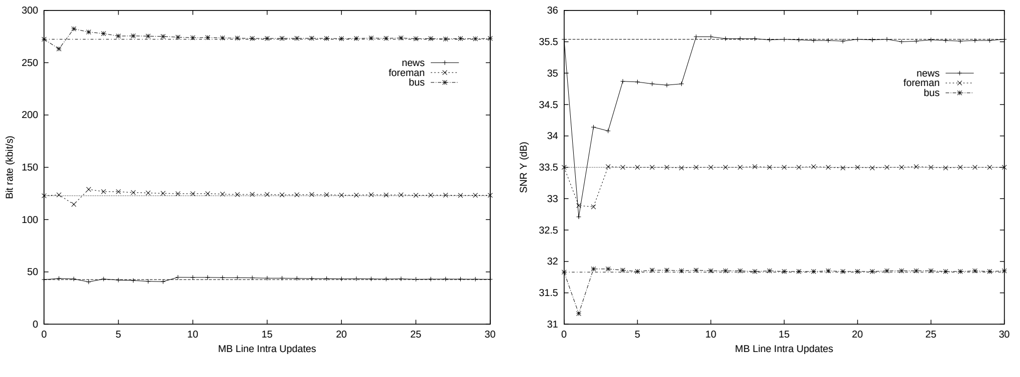


Figure 4.10: Bit rate and PSNR results for loop control with variable number of Macroblock Intra Updates

As shown in figures 4.11 and 4.12, the results obtained are quite similar to the use of intra coded frames: higher updating frequency can lead to improved error resilience, specially on scenarios that provoke a high number of packet losses.

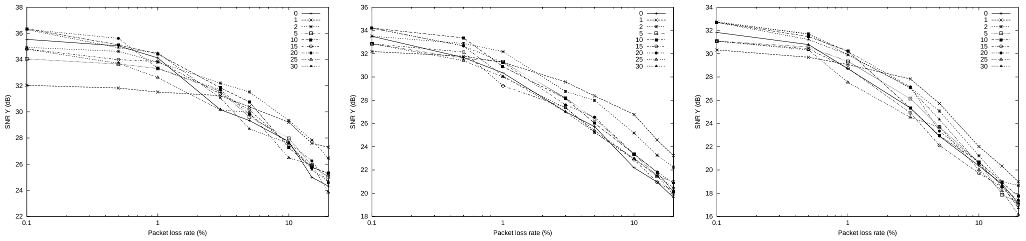


Figure 4.11: PSNR results with packet losses for News (left), Foreman (center) and Bus (right) sequences.

Concerning robustness we can generally say that, as happened with I frames, more frequent line intra updates result in higher robustness.

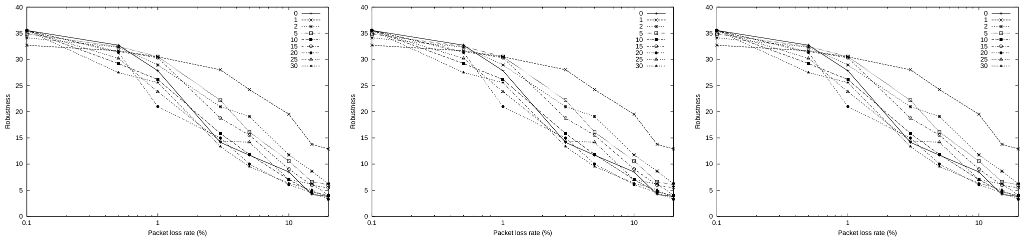


Figure 4.12: Robustness results for News (left), Foreman (center) and Bus (right) sequences.

4.3.5 Multiple reference frames

The use of multiple reference frames has two main purposes: increase compression by improving the prediction process and to increase error-resilience by offering a method that attempts to bypass situations where a previous frame cannot be used for reference since it was lost. The price to pay is the increased complexity and buffering at both encoder and decoder.

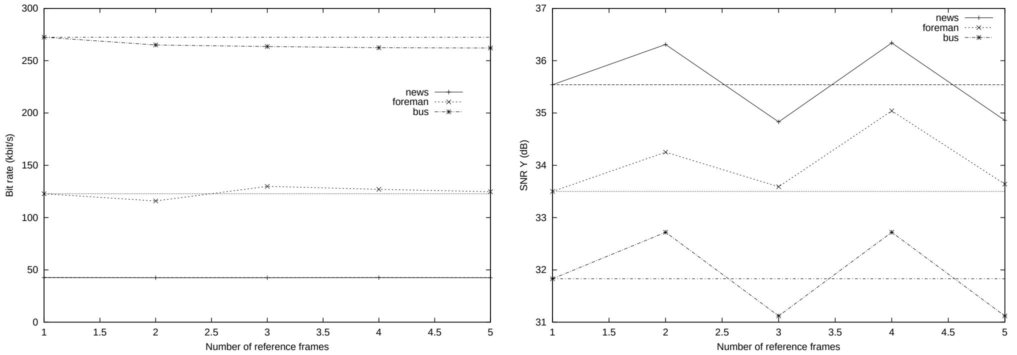


Figure 4.13: Bit rate and PSNR results for loop control with variable number of reference frames

Figure 4.13 shows strange effects on the distortion output of the rate controller. Has can be seen, the bit-rate control process has almost perfect results, with the distortion showing oscillations. This can be due to the number of bits used to code the number of the reference frames by using Exp-Golomb bit strings.

As it can be seen in figures 4.14 and 4.15, there are differences in distortion by using this technique, but its error-resilience and robustness features are not so evident. In fact, it can be noticed that the use of multiple reference frames can at times provide better behavior on error-prone scenarios, but it is not clear what value is best.

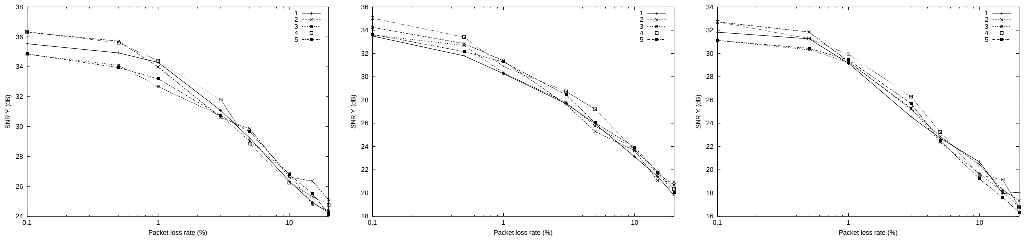


Figure 4.14: PSNR results with packet losses for News (left), Foreman (center) and Bus (right) sequences.

In terms of robustness this distinction is unclear too. This can be due to the fact that although there are frequent losses, these losses do not result in the loss of entire frames. This will explain why this method does not show great success.

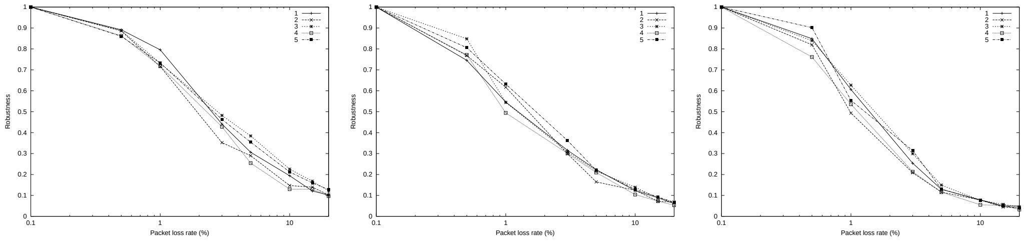


Figure 4.15: Robustness results for News (left), Foreman (center) and Bus (right) sequences.

4.3.6 Number of SP frames

The use of SP frames for error-resilience features shows a similar behavior to other types of intra coding presented previously. Even though their use causes a drop in quality for the same bit rate, they are used mostly to allow quick searches in the video stream and offer other features as well.

The results presented in figures 4.17 and 4.18 show that this kind of frames can be efficiently used in error-prone environments. However, using SP frames for increasing error-resilience performance is not the best solution, as we will show in the next point.

4.3.7 Random Intra macroblock refresh

This technique offers an excellent method to cope with the most demanding scenarios in terms of error-resilience since it presents a behavior of distributed intra updating. Such behavior removes the need for

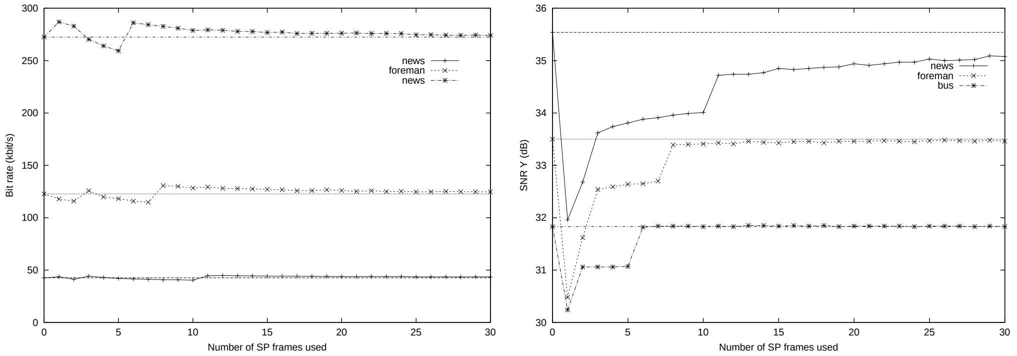


Figure 4.16: Bit rate and PSNR results for loop control with variable number of SP frames

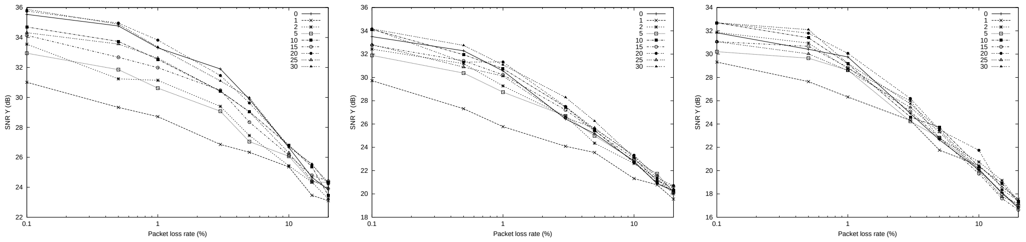


Figure 4.17: PSNR results with SP frames for News (left), Foreman (center) and Bus (right) sequences.

intra coded frames, and moreover it offers a simple method to approach a constant bit rate in the video stream.

Figure 4.20 shows that careful tuning of this parameter can achieve good results in scenarios with packet losses between 5% and 20%, which are acceptable values for wireless ad-hoc networks, for example. Intra-updating 1/3 of each frame offers a good balance between distortion and error-resilience, and can be considered as a good rule in such scenarios.

In terms of robustness, the obtained conclusions are similar to those achieved with intra frames: higher intra updating rates lead to higher robustness, since the capability of sustaining quality will be improved.

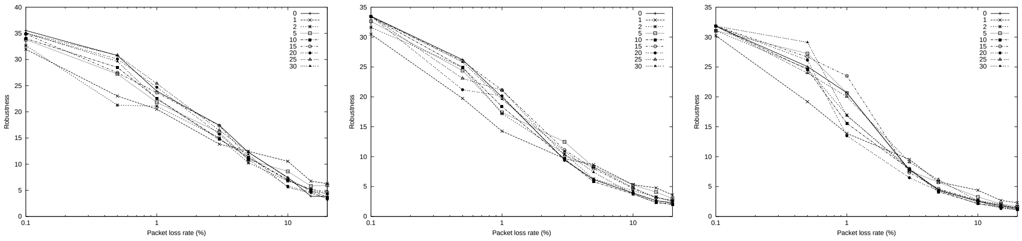


Figure 4.18: Robustness results with SP frames for News (left), Foreman (center) and Bus (right) sequences.

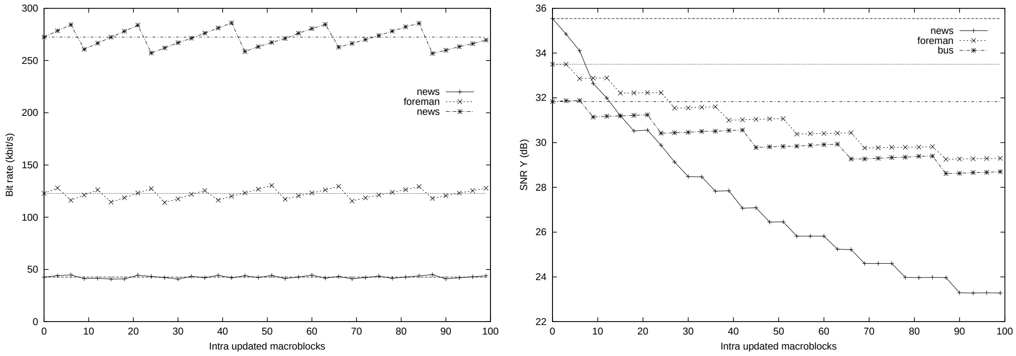


Figure 4.19: Bit rate and PSNR results for loop control with variable number of random intra macroblock refreshes

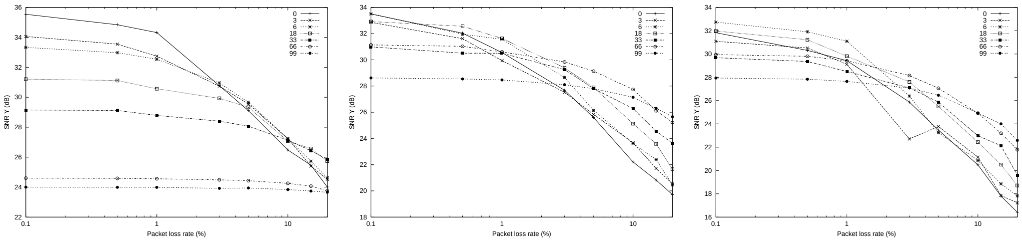


Figure 4.20: PSNR results with packet losses for News (left), Foreman (center) and Bus (right) sequences.

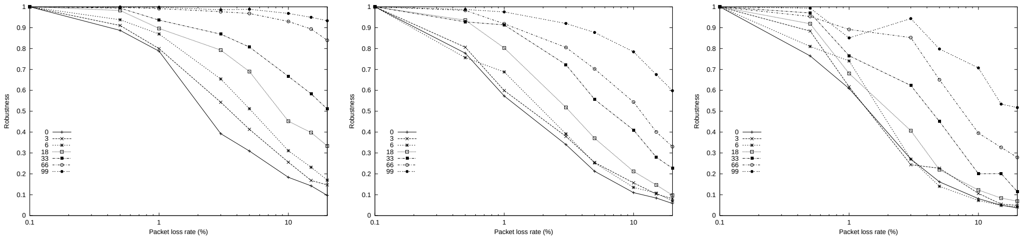


Figure 4.21: Robustness results for News (left), Foreman (center) and Bus (right) sequences.

Chapter 5

Evaluation of H.264 in bursty packet loss scenarios

We started our evaluation with small bursts affecting a single frame. Along with the regular decoding results, we also present the ones achieved by using Flexible Macroblock Ordering (FMO) mechanisms. This mechanism does macroblock rearrangement in order to distribute the burst error throughout the frame. Due to problems with the reference software, FMO tasks were performed by external software developed by the authors. Our evaluation was done using two and three groups of macroblocks; such options also belong to the H.264 framework. Other mechanisms, such as multiple reference frames, can also be used to cope with error bursts, so their evaluation was also included.

5.1 Bit rate vs. Packets/Frame

When streaming video through a network there is always a need of data packetization. This process can be tuned in order to obtain optimal performance in terms of throughput and error-resilience.

From the error-resilience point of view, fine-grain packetization will be optimal since the lost of a packet translated in the loss of just a small amount of data. From the network point of view, though, fine-grain packetization means higher bit-rate and so is prone to increase congestion.

Figure 5.1 shows the trade off between packetization and bit-rate. The bit-rate is calculated considering the actual 802.11b level 2 data, including the overhead introduced by RTP, UDP, IP and 802.11b itself. As it can be seen, the curves are not linear presenting a peak at some point. This is due to the relation between the number of macroblocks in a frame and the packetization process itself.

5.2 Effect of macroblock reordering on packet loss bursts

Macroblock reordering is a new feature in the H.264 codec that provides increased error-resilience when packet loss bursts appear. It aids the macroblock concealing process by spreading the error throughout the affected frame, reducing the temporal propagation of errors.

In this evaluation we used the Bus sequence, since it presents a high degree of movement. We considered that for that reason it properly stresses the codec for the evaluation being made. For our tests we used

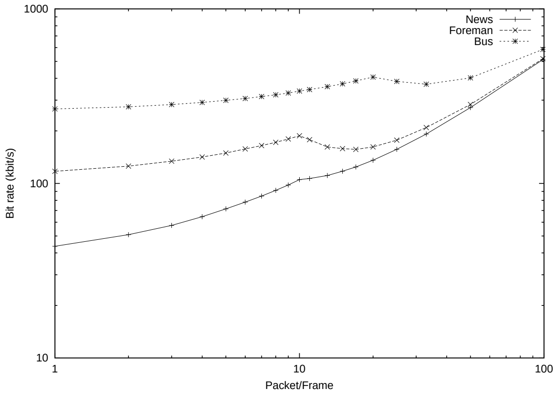


Figure 5.1: 802.11b bit rate vs. Packets/Frame for the three test sequences.

the first 30 frames and tuned the quantization to mid-scale for a distortion of 31.73 dB at 10 Hz.

The burst simulated occurs on the first P frame and was set to a duration of one quarter of frame, half frame and three quarters of frame.

Concerning the macroblock reordering process, the used reference software (JM3.9a) has that option broken, which led us to evaluate it by performing an external reordering of macroblocks. Unfortunately, the decoder crashed when a one macroblock per packet mapping was made, which forced us to use a minimum granularity of 2 macroblocks for testing.

Relatively to the reordering strategy, we divided the image in two groups (first the even, then the odd) and in three groups (first macroblocks 1, 4, 7, etc., then 2, 5, 8, etc., and finally 3, 6, 9, etc.). The H.264 reference software is expected to offer more complex reordering (such as spiral pattern distribution) in future implementations.

5.2.1 Loss burst of half frame

5.2.1.1 Even / odd scanning

As shown in table 5.1, FMO reordering achieves an average distortion increase of 1,6 dB, which is 7,6% higher than the solution with no FMO.

Robustness always shows an improvement by using FMO, though distortion itself presents worse results when the error occurs at the beginning of the frame.

Burst start relative to frame	PSNR	PSNR (FMO)	Robustness	Robustness (FMO)
Beginning of frame	27.70	27.52	0.201	0.227
Middle of frame	26.22	28.70	0.175	0.346
End of frame	26.12	28.62	0.178	0.276
Average Value	26,68	28,28	0.185	0.283

Table 5.1: PSNR and robustness average results for bi-partitioning after the loss of half frame.

This is due to the levels of movement in the image, which are lower on top and higher on the middle and bottom.

In relation to the initial distortion value for this sequence, we can see that on average the use of FMO reordering closes the gap towards that value by 31,68%, showing its effectiveness in relation to the best results that could be achieved.

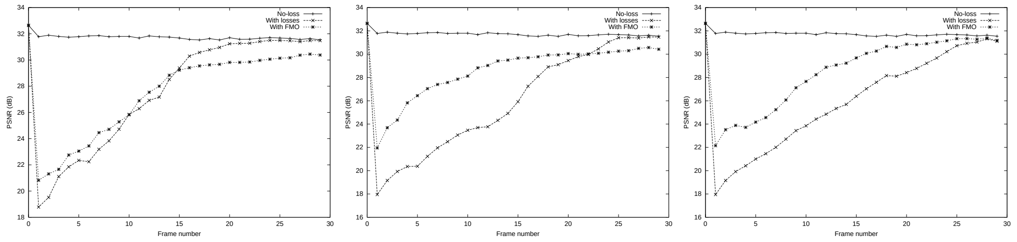


Figure 5.2: PSNR recovery with 2 groups after half-frame loss on the beginning (left), middle (center) and end (right) of frame 1.

Analysis of figure 5.2 shows that on the long term the spreading of errors by FMO reordering can lead to higher converging times. Since the solution with FMO always produces improved results before frame 16, it gives us a hint relative to minimum intra-updating frequency, which should be at least every 15 frames.

As expected, the highest distortion benefit is normally present on the frame which suffers the loss, validating FMO usefulness completely.

Figures 5.3, 5.4 and 5.5 show what occurs when the loss burst occurs at the beginning, middle and end of frame respectively. The error presented in those pictures was obtained through subtraction of the correspondent lossless frames; afterwards, the color levels were inverted so that white pixels mean that no error exists. The effect of using FMO on the first affected frame is clearly evidenced on these figures, being the error spread throughout the frame as expected. This behavior helps error-concealment techniques at the decoder side, showing better rate-distortion performance.

After 8 frames we can see that the error propagation has been reduced by using the FMO technique, being almost residual.

5.2.1.2 Triple scanning

We now present the results achieved by dividing the image in 3 macroblock groups instead of 2.

The performance is slightly inferior to the 2 groups solution. Table 5.2 presents the results achieved.

Burst start relative to frame	PSNR	PSNR (FMO-3)	Robustness	Robustness (FMO-3)
Beginning of frame	27.70	27.06	0.201	0.210
Middle of frame	26.22	27.77	0.175	0.254
End of frame	26.12	29.03	0.178	0.302
Average Value	26.68	27.95	0.185	0.255

Table 5.2: PSNR and robustness results for tri-partitioning with half frame loss.

As can be seen from that table, even though this method leads to inferior results, there is still an average improvement of 1,27 dB relative to the original solution. That corresponds to an increase of 4,76%, and the gap towards the maximum value is closed by 25,15% (6% less than the 2 groups solution).

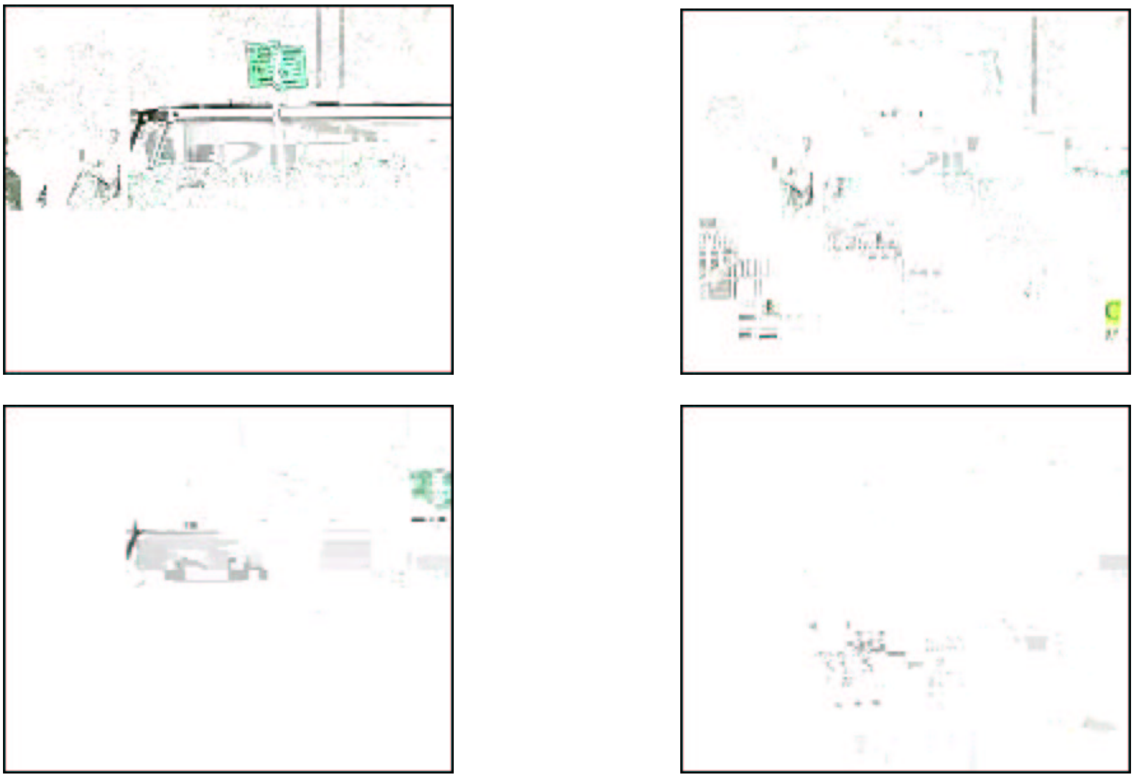


Figure 5.3: Error with (right) and without (left) macroblock reordering for burst at the beginning of the frame: on affected frame (top), after 8 frames (bottom)

5.2.2 Loss burst of 1/4 frame

This evaluation is used to compare the results obtained in the half-frame loss evaluation in order to check the generality of the conclusions achieved.

5.2.2.1 Even / odd scanning

By forcing quarter-frame losses, the drop in distortion is not as severe as was previously. The use of FMO reordering, therefore, still brings benefits, as it can be seen in table 5.3.

Burst start relative to frame	PSNR	PSNR (FMO)	Robustness	Robustness (FMO)
Beginning of frame	30.40	29.51	0.498	0.334
1/4 of frame	27.91	28.87	0.230	0.375
1/2 of frame	28.25	30.44	0.268	0.532
3/4 of frame	29.15	29.23	0.348	0.347
Average Value	28.93	29.51	0.336	0.397

Table 5.3: PSNR and robustness results after quarter-frame loss with macroblock bi-partitioning

The average increase in terms of distortion is of 0,57 dB (2%). This closes 20,7% of the gap towards the maximum value, which in terms of performance is not so good as with half-frame losses.

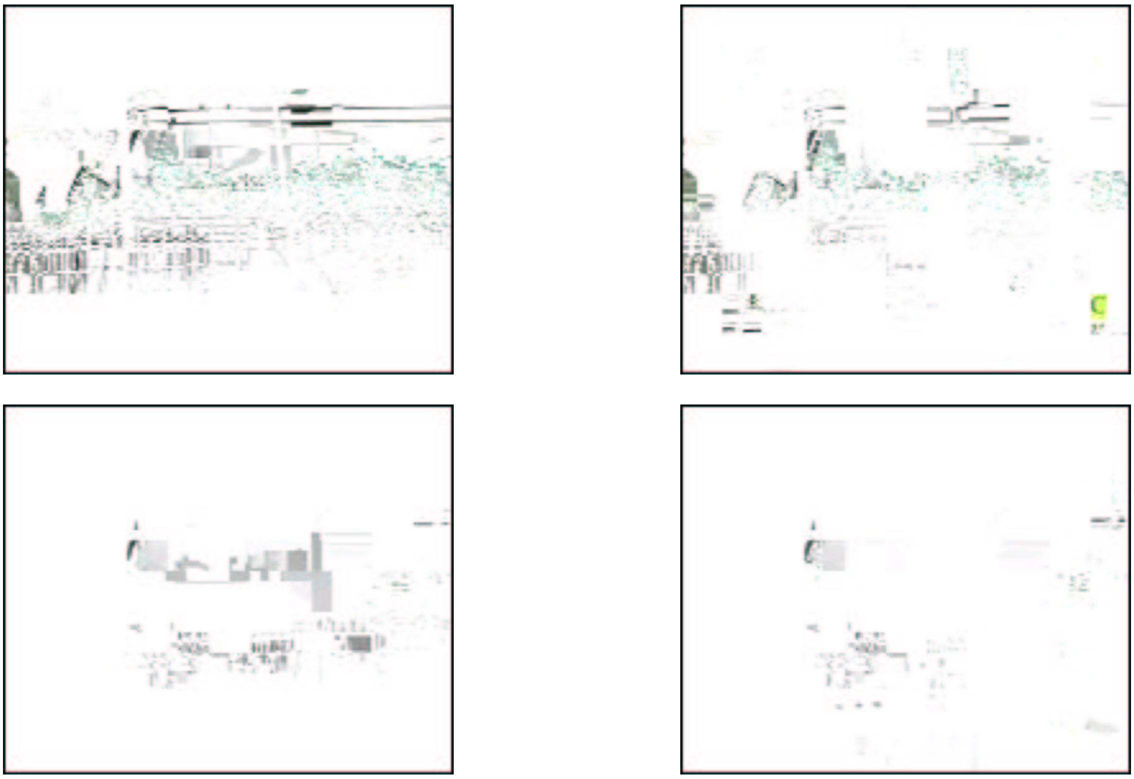


Figure 5.4: Error with (right) and without (left) macroblock reordering for burst in the middle of the frame: on affected frame (top), after 8 frames (bottom)

If we observe figure 5.7 we can see that when the loss occurs at the beginning of the frame (upper left picture), using FMO will produce much worse results (FMO curve is below the one with no FMO reordering). The upper quarter of the picture much more static than the rest. We can also see that when the error burst occurs at the end of the frame there is only slight gain for the same reason too. For losses occurring in the rest of the picture FMO produces better results except, as happened previously, after several frames because of difficulty to stop the error-propagation.

5.2.2.2 Triple scanning

Performing a triple scan continues to result in worse performance compared to a dual scan for macroblock reordering.

Table 5.4 shows the actual values. Now the average increase is of just 0,49 dB (1,7%), with 17,5% of the gap towards the optimum value being closed.

The results presented in figure 5.8 allow us to reach similar conclusions to those with frame bi-partitioning. Again, the loss of the first quarter of the frame is the only situation where FMO produces worse results. Error-propagation phenomena occurs in this situation too, and again frame 15 seems to be a transition point between applying FMO or not, as it can be seen in the picture at the upper right.

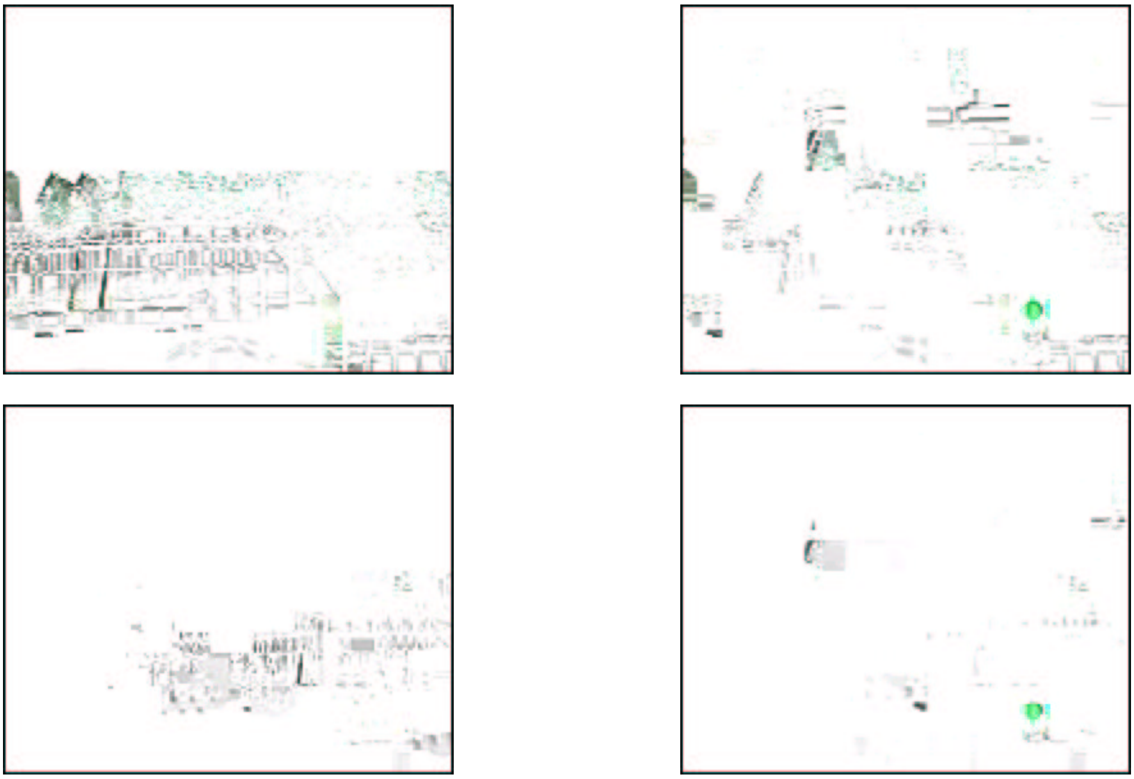


Figure 5.5: Error with (right) and without (left) macroblock reordering for burst in the end of the frame: on affected frame (top), after 8 frames (bottom)

5.2.3 Loss burst of 3/4 of frame

Previously we presented the results of quarter frames losses to perform a comparison towards the reference evaluation - half frame loss. We now do so with 3/4 frame losses to see what happens for burst longer than those used for the reference situation.

5.2.3.1 Even / odd scanning

Table 5.5 shows that on average FMO presents worse distortion values by a difference of 0,26 dB. Robustness, though, still shows an increase.

Taking a look at figure 5.9 we can see that what is actually happening is that FMO still provides superior results, though only slightly at times. After several frames FMO reordering can show a much worse behavior. This explains, therefore, the worse average results reached.

5.2.3.2 Triple scanning

Contrarily to dual scanning, triple scanning on the FMO process does produce improved average results, as exposed in table 5.6. Now the increase is on average of 1,21 dB (4,63%), and 21,49% of the gap towards the optimum value is closed.

This result is quite unexpected, since the two scan option (bi-partitioning) is more effective at spreading errors.

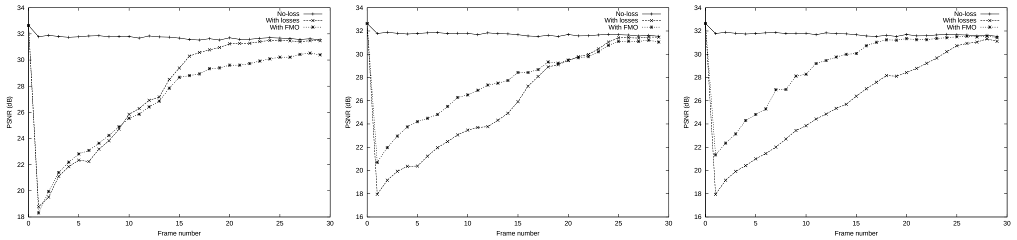


Figure 5.6: PSNR recovery with tri-partitioning after half-frame burst during the beginning (left), middle (center) and end (right) of frame.

Burst start relative to frame	PSNR	PSNR (FMO-3)	Robustness	Robustness (FMO-3)
Beginning of frame	30.40	28.74	0.498	0.284
1/4 of frame	27.91	28.45	0.230	0.324
1/2 of frame	28.25	30.77	0.268	0.628
3/4 of frame	29.15	29.70	0.348	0.368
Average Value	28.93	29.42	0.336	0.401

Table 5.4: PSNR and robustness results after a quarter-frame loss with macroblock tri-partitioning

As shown in figure 5.10, the improvement is mainly due to the performance achieved when the error occurs in the lower part of the frame, which in this case is the one with higher level of movement.

5.3 Multiple reference frames

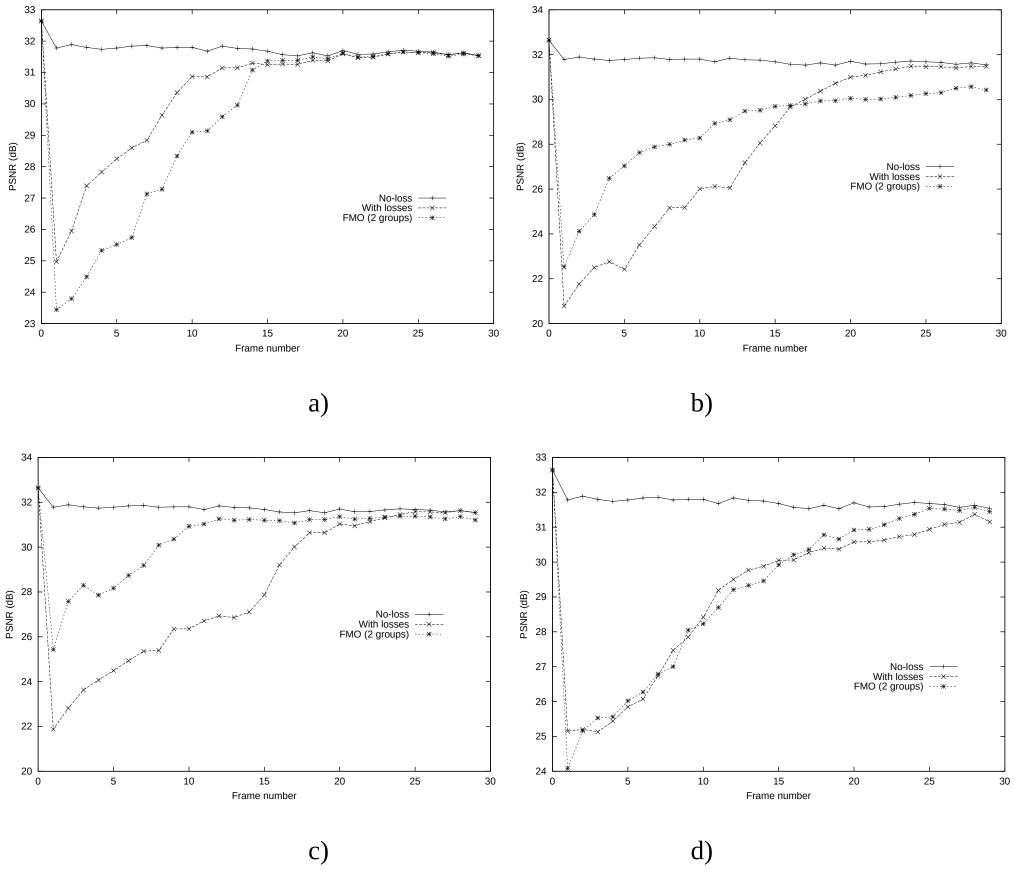
To evaluate the error-resilience properties of the multiple reference frames technique, we took the Foreman QCIF sequence and provoked losses ranging from 1 to 5 consecutive frames, so that the error propagation effect was presented as clearly as possible. The analysis was done using 1, 3 and 5 reference frames.

Figure 5.11 presents the results achieved. Number/arrow pairs refer to how many frames were lost.

As it can be seen, the behavior experienced is quite clear: more reference frames lead to worse results every time.

This result was unexpected according to the works [8, 9], which state that the use of multiple frames of reference are effective in terms of error-resilience. From our results, though, we can conclude that using a single reference frame is the most effective choice to stop temporal error propagation. Demands in terms of memory on both encoder and decoder are also reduced by this setup.

We are not aware if the result we obtained is due to an error in the implementation of H.264 codec, though we believe not.



a) b)

c) d)

Figure 5.7: PSNR recovery with bi-partitioning after quarter-frame loss at a) beginning of frame b) 1/4 of frame c) half frame d) 3/4 of frame.

Burst start relative to frame	PSNR	PSNR (FMO)	Robustness	Robustness (FMO)
Beginning of frame	26.61	26.51	0.183	0.183
End of frame	25.58	25.17	0.158	0.172
Average Value	26.10	25.84	0.171	0.178

Table 5.5: PSNR and robustness results after loss of 3/4 of frame with macroblock bi-partitioning.

Burst start relative to frame	PSNR	PSNR (FMO-3)	Robustness	Robustness (FMO-3)
Beginning of frame	26.61	26.74	0.183	0.189
End of frame	25.58	27.88	0.158	0.230
Average Value	26.10	27.31	0.171	0.210

Table 5.6: PSNR and robustness results after loss of 3/4 of frame with macroblock tri-partitioning

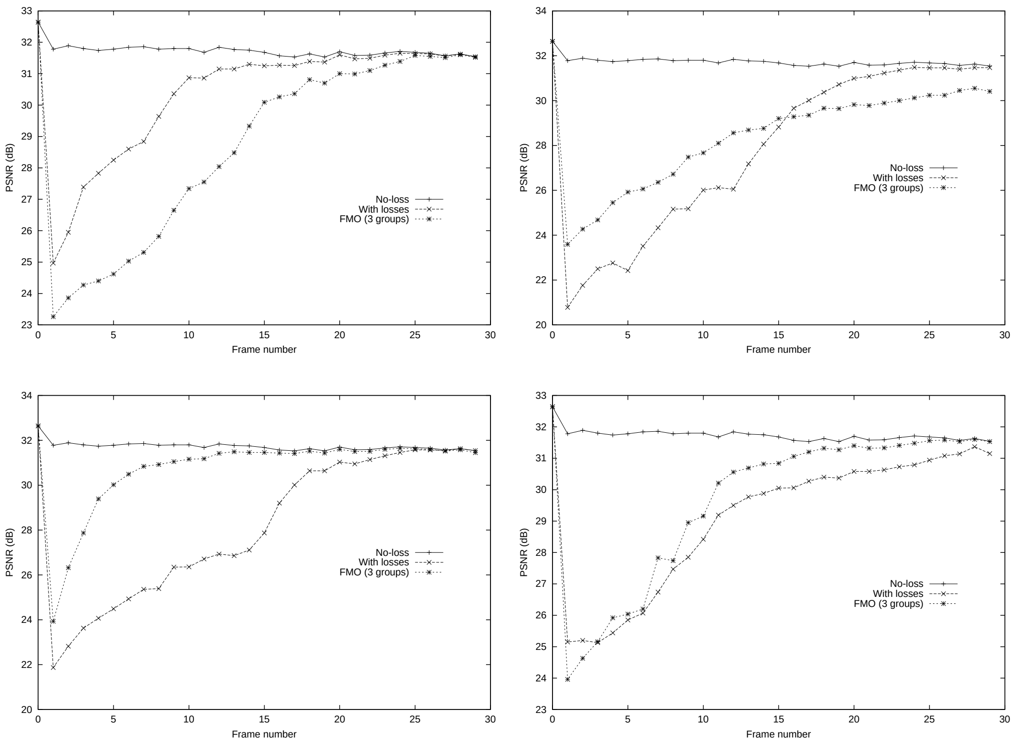


Figure 5.8: PSNR recovery with tri-partitioning after quarter-frame loss at different frame positions.

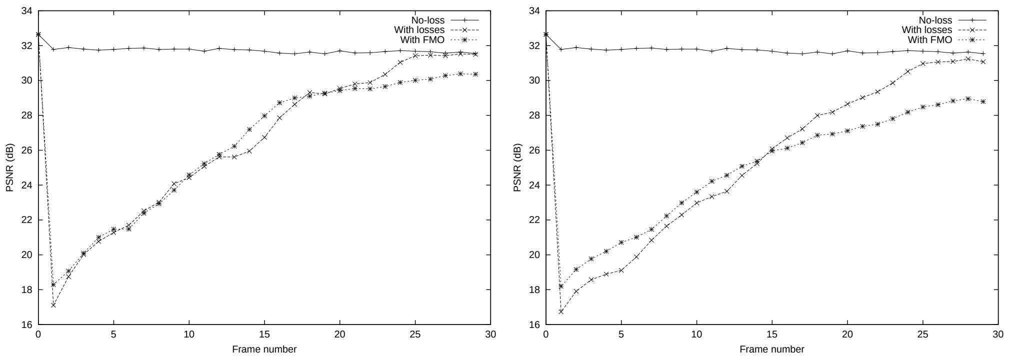


Figure 5.9: PSNR Recovery after burst during the beginning and end of frame 1.

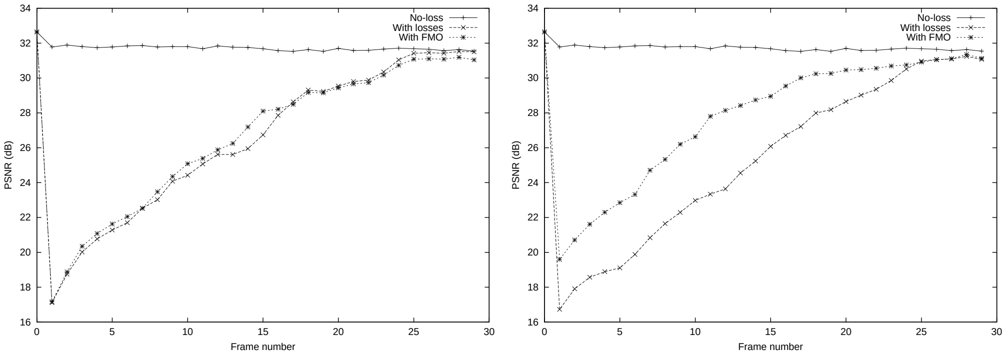


Figure 5.10: PSNR recovery with tri-partitioning after loss burst during the beginning (left) and end (right) of the frame.

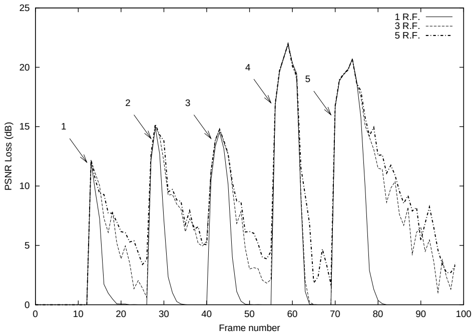


Figure 5.11: Analysis of error propagation by simulating an increasing number of entirely lost frames

Chapter 6

Conclusions

In this report we present some of the performance results obtained from our evaluation of the H.264 codec.

We have analyzed the behavior of the various tools included, presenting the results in terms of distortion, bit-rate and processing overhead. Using those results we tuned the codec in order to achieve optimum performance in terms of compression, and then we proceeded to analyze the effectiveness of the error-resilience tools belonging to the H.264 framework.

Concerning the error resilience tests, our analysis was centered on two types of error: random and bursty packet losses. The random-loss results obtained allows tuning the encoder according to the expected packet loss rates inside the network, and show that a careful choice can increase significantly the overall PSNR of the sequence. We also present the effects of packet bursts on the quality of video and propose methods to efficiently handle these situations. Assuming a typical situation where there is a 10% packet loss in the network, tuning the Random Intra Macroblock Update to 1/3 of the frame size improves error-resilience on random and burst error situations at the cost of only a marginal increment in bit-rate.

Concerning the use of multiple reference frames, our study points out to the use of just two reference frames as a reasonable option facing random errors. Concerning error-bursts, it was shown that this technique increases the temporal error propagation and it should be avoided, expect for situations where the media is reliable (CD, DVD, or hard-disk).

The FMO technique will also perform significantly well on small bursts at a small cost in terms of processing time.

The results presented in this paper enable tuning the H.264 codec for different network scenarios appropriately. In later works we will study its behavior in more realistic situations, using the NS-2[11] tool to simulate the effects of the different ad-hoc routing protocols on video stream flows, addressing more detailed and appropriate models to radio transmission issues inherent to wireless networks.

Bibliography

[1] ITU Telecommunication Standardization Sector (ITU-T). http://www.itu.int/ITU-T.

[2] Joint Technical Committee 1. http://www.jtc1.org.

[3] Video coding for low bitrate communication. ITU-T Recommendation H.263, 1995.

[4] ISO/IEC IS,Coding of Audio-Visual Objects, part 2: Visual (MPEG-4). Information Technology, November 2001.

[5] MPEG Group Home page. http://mpeg.telecomitalialab.com.

[6] Thomas Stockhammer, Dimitrios Kontopodis, and Thomas Wiegand. Rate-Distortion Optimization for H.26L Video Coding in Packet Loss Environment. *12th International Packet Video Workshop (PV 2002), Pittsburg, PY*, May 2002.

[7] Thomas Stockhammer and Stephan Wenger. Standard compliant enhancements of jvt coded video over fixed and wireless ip. *2002 International Tyrrhenian Workshop on Digital Communications (IWDC 2002), Capri (Italy)*, September 2002.

[8] M. Budagavi and J. D. Gibson. Error Propagation in Motion Compensated Video over Wireless Channels. *Proceedings of the IEEE International Conference on Image Processing, Santa Barbara, USA*, pages 89–92, October 1997.

[9] M. Budagavi and J. D. Gibson. Multiframe video coding for improved performance over wireless channels. *IEEE Transactions on Image Processing, Volume: 10, Issue: 2*, pages 252–265, February 2001.

[10] Ye-Kui Wang, Miska Hannuksela, Viktor Varsa, Ari Hourunranta, and Moncef Gabbouj. The error concealment feature in the h.26l test model. *IEEE 2002 International Conference on Image Processing (ICIP'2002), Rochester, New York, USA*, September 2002.

[11] K. Fall and K. Varadhan. Ns notes and documentation. The VINT project, UC Berkeley, LBL USC/ISI, and Xerox PARC, November 1997.