Chapter 3

Error Resilience of H.264 + RFGS

In this thesis, we adopt the H.264 + RFGS structure as the base for the discussion of error resilience techniques. In this chapter, we will demonstrate how the performance of error resilience is influenced by some parameters and modes in the H.264 + RFGS scheme. These parameters that will be taken into account include the length of packets (PL), the Quantization Parameter (QP), the leaky factor (α) and the partial prediction parameter (β) in RFGS. In addition, packaging modes, including FMO and field coding, are to be especially discussed.

3.1 Effects of Packet Length

Decision of packet length is a general issue in video streaming no matter which video coding structure is used. In visual communications, video data are usually packaged into packets before they are put into a network. Hence, as we discuss the data error problem, we are actually dealing with the packet loss problems.

There are two major sources for packet loss in data communications: (1) bit errors caused by noise in physical channels and (2) packet errors occurring in packet-switch networks. Both types of errors are relevant to packet length. When transmitting data in a noisy network, the shorter the packet length is, the lower the packet loss rate (PLR) will be. On the other hand, shorter packets bring a large number of overheads and thus reduce coding efficiency. Hence, there would be a trade-off between transmission quality and coding efficiency. A way to decide an acceptable choice of packet length is to minimize the cost function (C), which is defined as

$$C = PLR + \lambda\, dR. \tag{3.1}$$

Here PLR denotes the packet loss rate and R represents the coding rate. λ would be set larger when we ask for better coding efficiency. Otherwise, λ would be set smaller if we aim for lower packet loss rate.

Let the PLR caused by bit errors be represented as PLR_b (BER, PL) while the PLR caused by packet errors be represented as PLR_p (PL). Here, PLR_b would be a function of BER and PL (Packet Length), with BER being the bit error probability in the physical channels and PL the packet length. On the other hand, PLR_p would be a function of PL. With PLR_b and PLR_p, the total effective packet loss rate could be formulated as

$$PLR_{effective} = PLR_p\,(PL) + (1- PLR_p\,(PL))\; PLR_b\,(BER, PL) \tag{3.2}$$

The analysis of PLR_p depends on many factors, like the traffic of networks and the buffer size of routers. To simplify the problem, we assume PLR_p is known empirically and only focus on the discussion of PLR_b (BER, PL).

Assume the effective bit error probability is denoted as P. Here we assume channel coding is applied and bit-correction has already been performed at the application layer. That is, P means the "effective" bit error probability after error correction. It is supposed that after error correction the existence of any erroneous bit causes a packet loss. With this assumption, the PLR_b is estimated to be

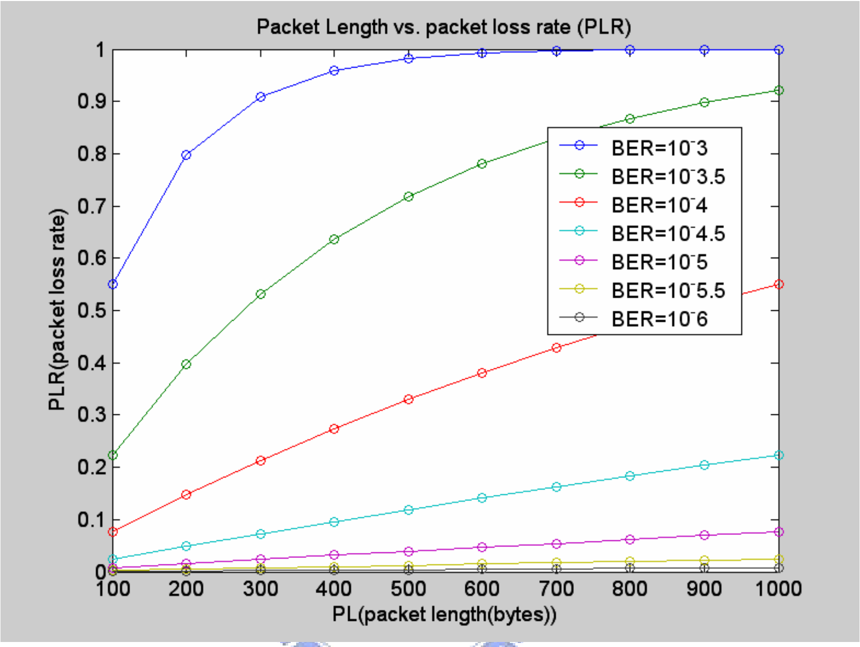
$$PLR_b\,(BER, PL) \approx 1 - (1\text{-}P) \wedge (X*8). \tag{3.3}$$

Fig. 3.1(a) shows the curves of Eq.(3.3). It is clear that PLR_b is increasing as PL is getting larger.

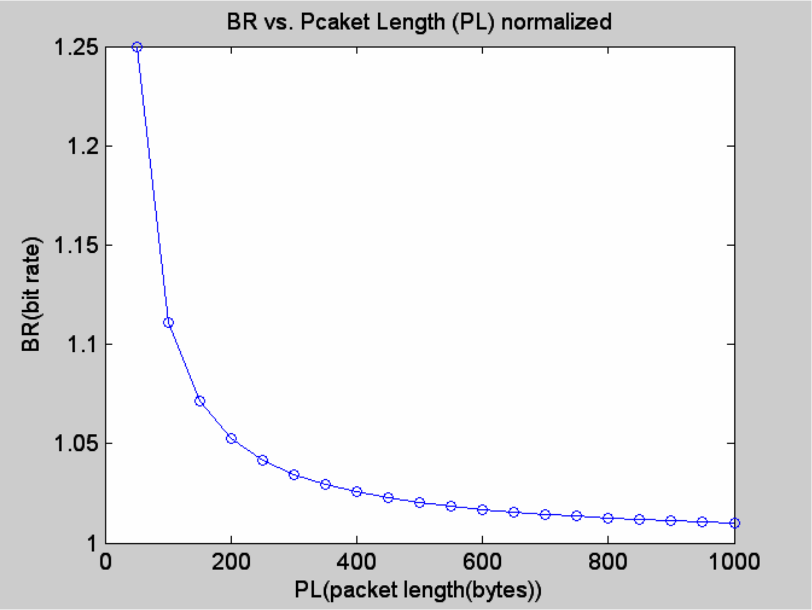
In H.264, the header bytes per packet approximate 10 bytes. We can define a coding rate ratio R as formulated below

$$R=PL/(PL\text{-}10)\,. \tag{3.4}$$

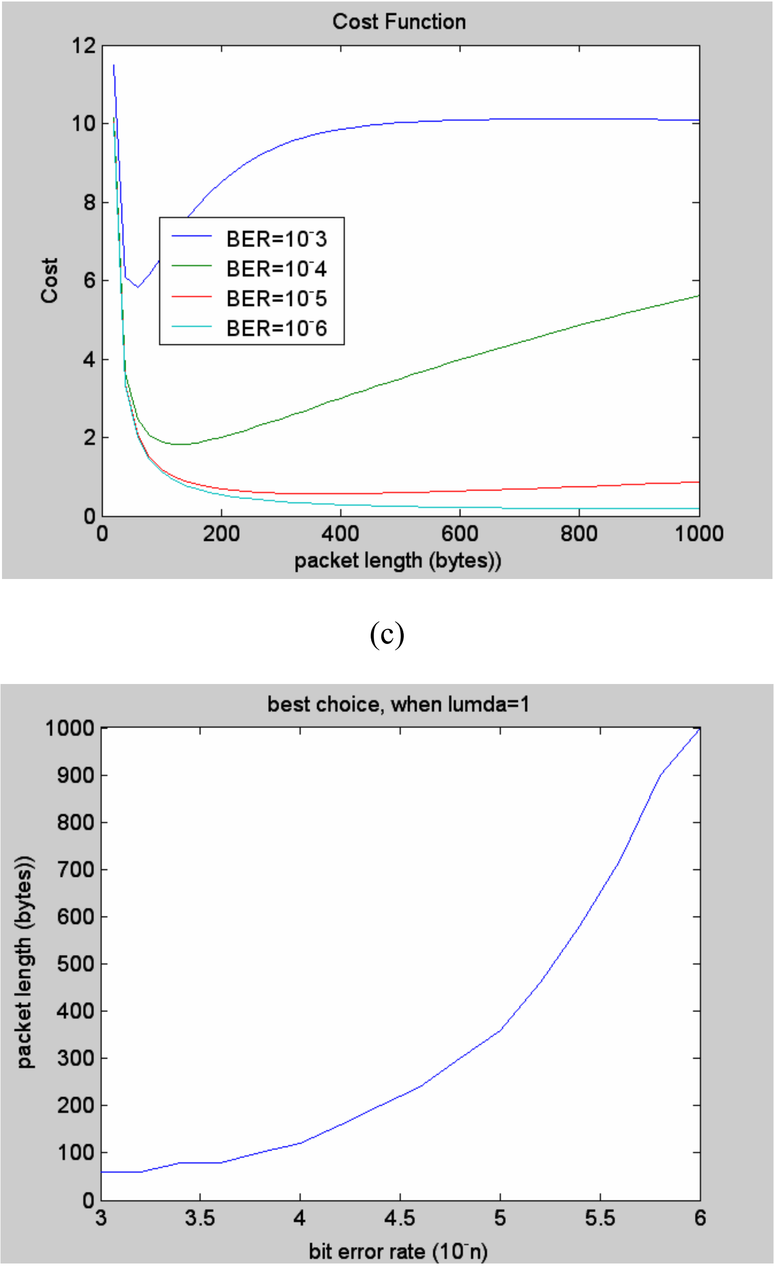
The curve of Eq.(3.4) is shown in Fig. 3.1(b). Now we may substitute Eqs.(3.3) and (3.4) into Eq.(3.1) and choose appropriate values of λ and BER to fit our requirement. By minimizing C, the corresponding PL can be gotten. Fig. 3.1(c) shows the curves of the gain function (Eq.(3.1)) and Fig 3.1(d) gives the optimal packet length at different BER's when λ is set to be 1.



(a)



(b)



(c)

(d)

Fig. 3.1 Discussion of packet length: (a) packet loss rate versus packet length (PL) at different BER's; (b) normalized bit rate versus PL, assuming bit rate = 1 when there is no packetization; (c) curves of C with λ=1; and (d) optimal packet length at different BER's, with λ=1.

In Fig. 3.1(b), we can see a quick raise of bit rate when PL is decreased down to below 200 bytes. Hence, in later discussions, we will fix PL as 200 bytes for its relative low PLR_b and small coding rate ratio R. Moreover, the range of PLR_b is set to be 0~0.05.

3.2 Quantization Parameter (QP)

After the discussion of packet length, the next thing to discuss is about some specific parameters in H.264+RFGS that are relative to error resilience. Among these parameters, we discuss first the quantization parameter (QP).

QP is a parameter in H.264/AVC that decides the quantization step size of transform coefficients. A larger value of QP means a larger quantization step size. This results in a shorter code stream that could be transmitted over a low bit rate channel. However, the reconstructed video quality at the decoder side will be lower. On the contrary, coding with a small value of QP would generate videos of better quality, but the code stream would be longer and a channel of wider bandwidth would be needed. Since there is trade-off between bit rate and video quality, we usually select a QP to achieve the best affordable quality with the bit rate under the channel bandwidth constraint.

Fig. 3.2 shows a simulation result of the error resilience performance of the (non-scalable) single-layer H.264 coding scheme. In this simulation, the packet loss rate (PLR) is about 1%. Since a single packet may contain the information of several macroblocks, the loss of a single packet may cause all the information of several macroblocks to be lost, including motion vectors, DCT coefficients, etc. Once several macroblocks of a frame are lost, the quality of that frame would be seriously declined. Moreover, all the successive frames would also be affected due to the effect of error propagation. For a larger QP, each packet contains more macroblocks and thus a packet loss results in the loss of more macroblocks. However, the decrease of base layer size makes the base layer less vulnerable to packet loss. The simulation result shows that a larger value of QP actually achieves better error resilience performance.

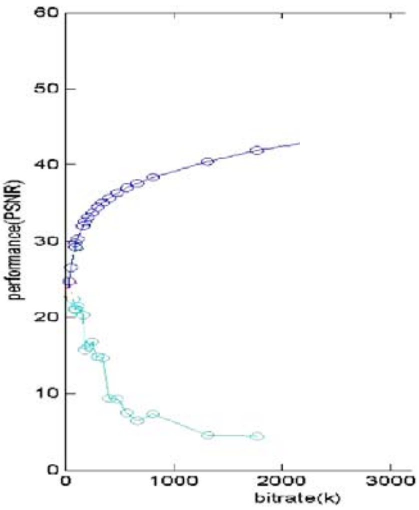
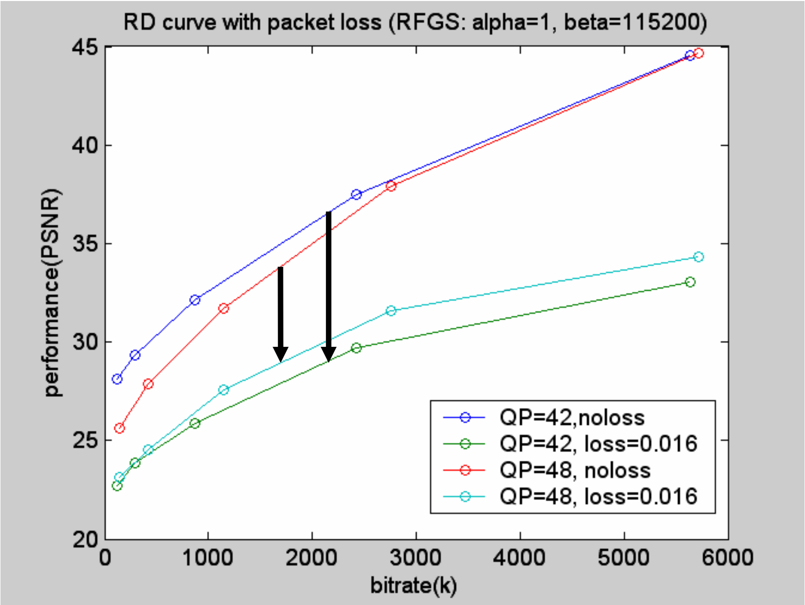


Fig. 3.2 Simulation results of error resilience performance of the single-layer H.264.

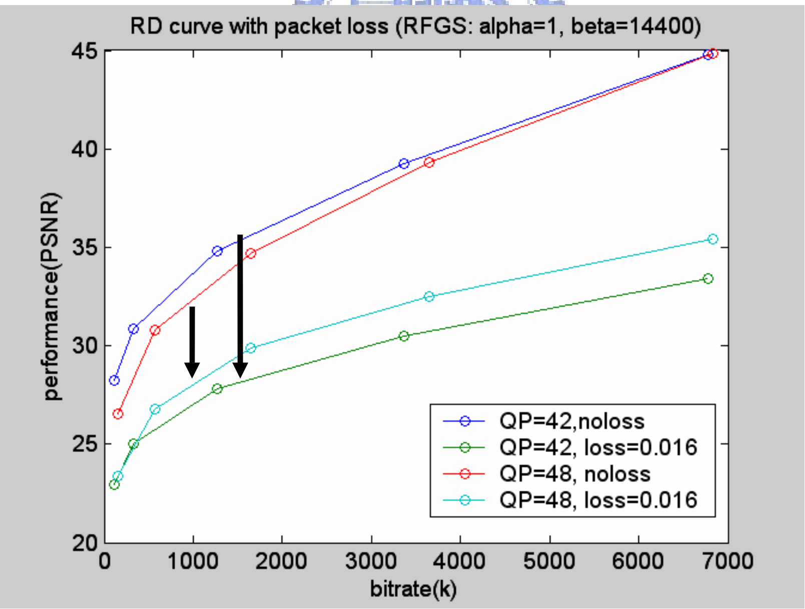
Packet length (PL) = 200 bytes, packet loss rate (PLR)=0.016
Blue line: No loss H.264 non-scalable coding
Light blue line: H.264 non-scalable coding, PLR = 0.016

QP also influences the performance of scalable coding. When applying RFGS on H.264/AVC, a video sequence is partitioned into base layer and enhancement layer. The base layer is coded with the H.264 coding scheme while the enhancement layer is coded with a bit-plane coding scheme. Since the scalable coding is less efficient than H.264 video coding, the amount of data partitioned into the base layer will affect the coding efficiency of H.264+RFGS. When the percentage of data in the base layer is raised, the coding efficiency is improved. Since in H.264 QP could control the size of based layer, we can say that the value of QP controls the coding efficiency of the scalable coding. A decreasing of QP value at the base layer tends to result in the increase of coding efficiency.

However, it doesn't mean the smallest value of QP would be the best choice. A major purpose of scalable video coding is to adapt to the variation of channel bandwidth. A small value of QP increases the lower bound of the required channel bandwidth and reduces the range of bit rate adaptation. Moreover, the choice of QP also influences the error resilience capability of scalable coding. In the H.264+RFGS scheme, a decreasing of QP would cause the descending of error resilience capability. This is because a higher percentage of data would be allocated into the base layer and thus there would be a higher probability of errors happening in the base layer.



(a)



(b)

Fig. 3.3 Simulation results of error resilience of H.264 +RFGS

(a) Enhancement reference bits = 115200 bits,

(b) Enhancement reference bits = 14400 bits

Blue line: No loss H.264+RFGS, QP=42

Red line: No loss H.264+RFGS, QP=48

Green line: H.264+RFGS, QP=42, PLR = 0.016

Light blue line: H.264+RFGS, QP=48, PLR= 0.016

In H.264+RFGS coding, the base layer plays a crucial role in the reconstructed visual quality. A packet loss happening in the base layer would affect the quality of video seriously. Hence, when the base layer is more error resilient, the overall error resilience of H.264+RFGS is usually better. Fig. 3.3 shows the simulation results of PSNR performance versus bit-rates for different values of QP. In Fig. 3.3 (a), more enhancement reference bits are used. We can see that a larger value of QP always results in a more error resilient code stream. The detail discussions of these enhancement reference bits are to be given in the next section.

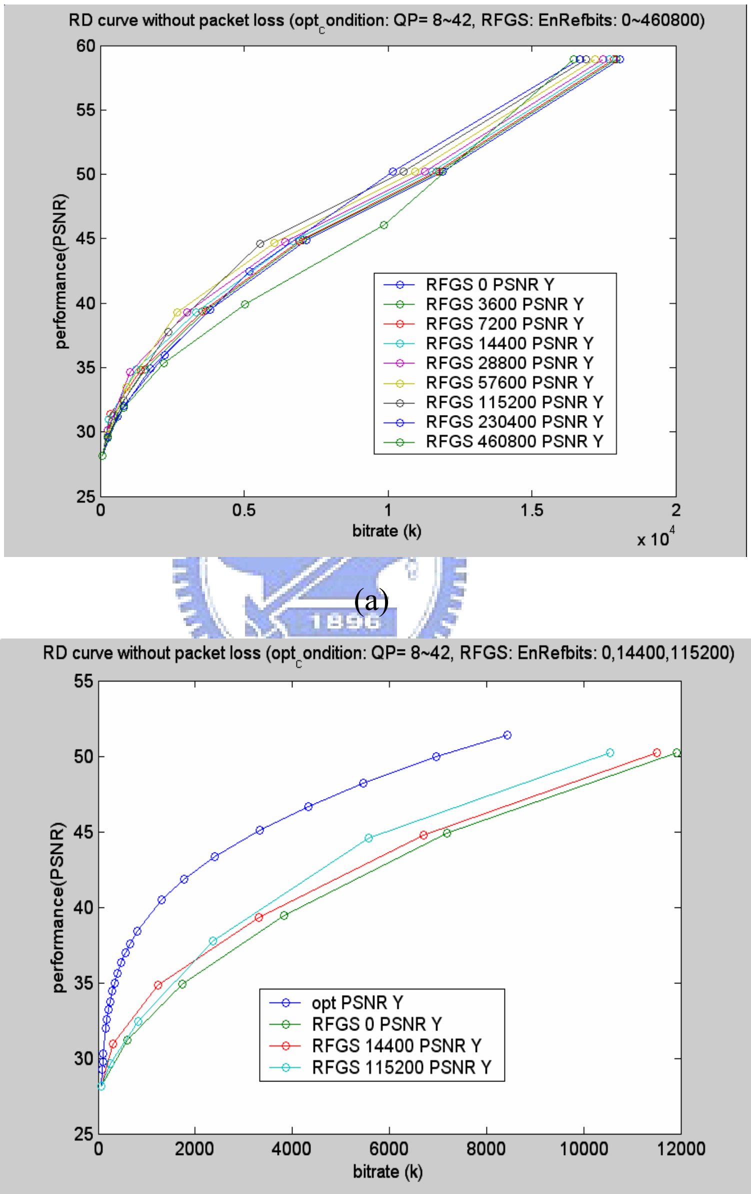
3.3 Partial Prediction Parameter (β)

β is a specific parameter in RFGS. β originally represents the number of bit planes used to form a higher-quality reference frame for the motion compensation of the next frame's enhancement layer. Here we adopt a modified definition of β which means the number of bits used to form the enhancement reference frame. With this definition, β is no longer limited to the number of bit-planes and we can have a continuous choice of β.

Fig. 3.4 shows the coding efficiency for different values of β. Here, β is changed from 0 to 460800 bits. Once a bit plane is decoded, a small circle is plotted on the figure to represent the bit rate and the luminance performance (i.e. PSNR Y) up to this bit plane. As the number of bit-planes increases, the number of coded bits increases drastically. Once the number of coded bits is large enough so that the PSNR value is over 45dB, the quality gain of the reconstructed video becomes indistinguishable to human eyes. In this case, there is no need to keep increasing the number of coded bits.

In Fig. 3.4(a), we can see that at a given bit rate, the video data are best coded with a specific choice of β. When the channel bandwidth is wider, a larger β can be used to achieve better coding efficiency. When the bandwidth is narrower, there would be no enough space for too many enhancement data and thus the use of a smaller β tends to be more efficient. As shown in Fig. 3.4(b), except the H.264 non-scalable coding, H.264+RFGS with a lower β has better coding efficiency when the bit rate is low. As the bit rate increases, the coding efficiency of H.264+RFGS with a larger β catches up and finally exceeds.

When there are packet losses, these bitstreams with better efficiency tend to perform poorer in terms of error resilience. Moreover, as shown in Fig. 3.5, as the packet loss rate gets higher, the PSNR performance of different β's become close to each other.



(a)

(b)

Fig. 3.4 The coding efficiency for different choices of β

(a) β = 0, 3600, 7200, 14400, 28800, 57600, 115200, 230400, 460800.

(b) Comparison among H.264 non-scalable coding, H.264+RFGS with zero β, H.264+RFGS with a small value β, and H.264+RFGS with a medium value of β.

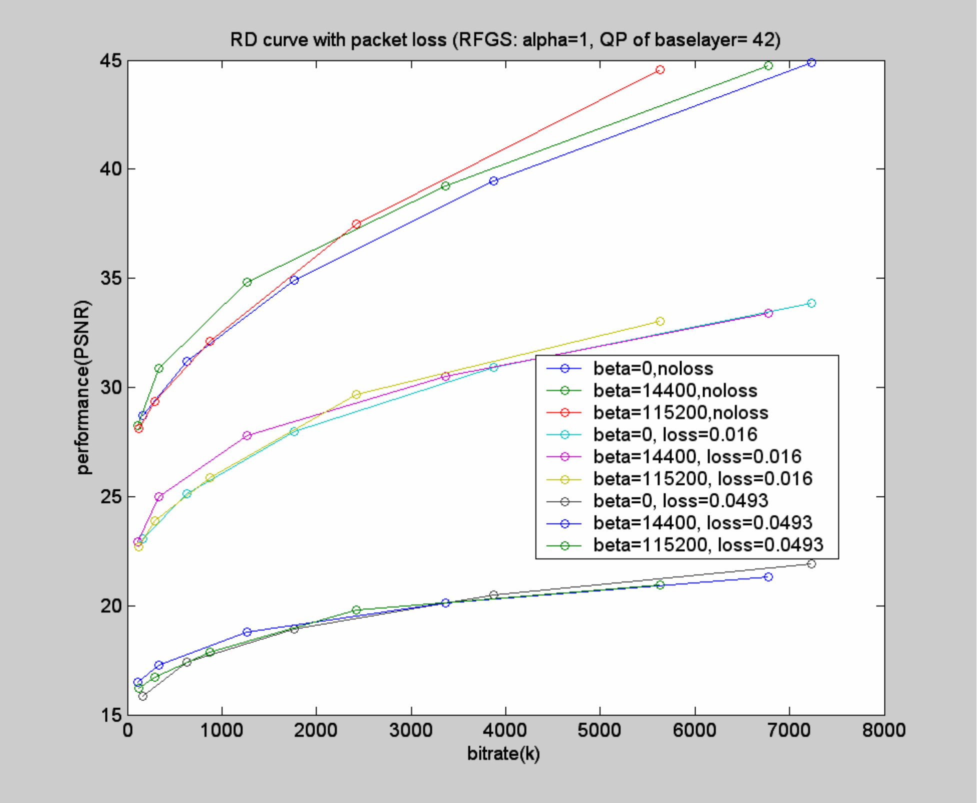


Fig. 3.5 PSNR performance versus bitrate with respect to various packet loss rates and different β's

3.4 Leaky Factor (α)

α is another specific parameter in RFGS. The leaky factor α controls the extent of error propagation. The use of a large α generates an efficient code stream but the bitstream becomes less resistant to error propagation. On the contrary, the use of a small α produces a less efficient code stream but the bitstream is better resistant to error propagation. Similar to Fig. 3.5, we show in Fig. 3.6 the impact of α over the PSNR versus bitrate curves. As the packet loss rate gets higher, bitstreams with better coding efficiency suffer the effect of error propagation more.

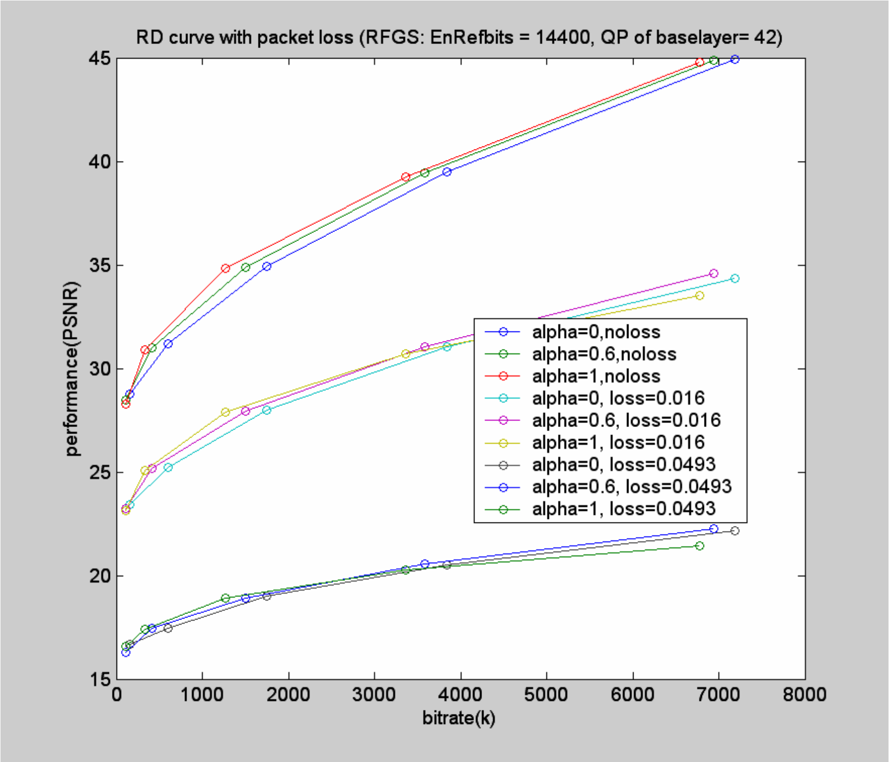


Fig. 3.6 PSNR performance versus bitrates with respect to various packet loss rates and different α's

3.5 Packaging Methods

In this section, we discuss how to package a frame to achieve better error resilience. When we try to improve the error resilience capability of H.264, two premises are expected:

1. Not to decrease coding efficiency too much. Since current channel coding techniques are already able to protect data with only a small amount of overhead, it is impractical to use an error resilience method that deteriorates too much the coding efficiency.
2. Not to modify the original H.264 coding standard too much. In this thesis, we aim to improve the error resilience capability of H.264/AVC with as less modification as possible.

In the H.264/AVC standard, there already exists a packaging method for the purpose of error resilience. This packaging method is called FMO (flexible macroblock ordering).

Fig. 3.8 shows the examples of FMO packaging. Here, each block represents a macroblock and blocks of different colors represent macroblocks of different groups. Each group of macroblocks is coded individually. Compared with the traditional sequential packaging way shown in Fig. 3.7, FMO is more error resilient. When packet losses happen, FMO can disperse the error and make the error concealment at the decoder side more efficient and effective. However, the use of FMO packaging produces too many overheads and thus decreases the coding efficiency. As shown in Table 3.1, the bit rate of H.264 coding with FMO packaging becomes 1.33 times of the original one. Hence, in this thesis we develop another packaging method, which is based on field coding, to achieve better error resilience with a reasonable trade-off of coding efficiency.

As shown in Fig. 3.9, field coding partitions a frame into two fields – the top field contains the even-numbered rows and the bottom field contains the odd-numbered rows. With this field structure, once some packets of a field are lost, the information of lost pixels could be easily recovered based on the information in the other field. The error concealment method would be very easy and efficient. On the contrary, with the FMO packaging approach, the loss of packets causes the loss of macroblocks and it would be much difficult to conceal these lost macroblocks.

However, the coding efficiency of the H.264 default structure of field coding is unstable in different video sequences. The coding efficiency is higher in a frame with uncomplicated content. A frame with more complicated content lowers the coding efficiency more. As shown in Table 3.1, in average the coding efficiency of default field coding is very low. Hence we manage to modify the structure of field coding to enhance its coding efficiency. As will be demonstrated in the next chapter, with modified structure, field coding only increases 3% the bit rate compared with the standard frame-based H.264 coding. This fact makes the proposed packaging scheme based on field coding even more attractive. All the details are to be described in the next chapter.

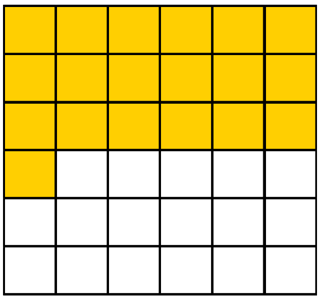


Fig. 3.7 The sequential packaging

(Each block represents a macroblock.)

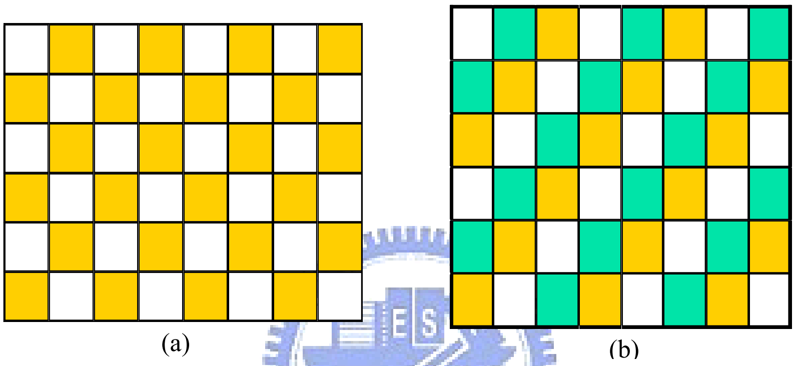


Fig. 3.8 Two types of FMO packaging

(a) Two-group packaging, (b) three-group packaging

(Each block represents a macroblock.)

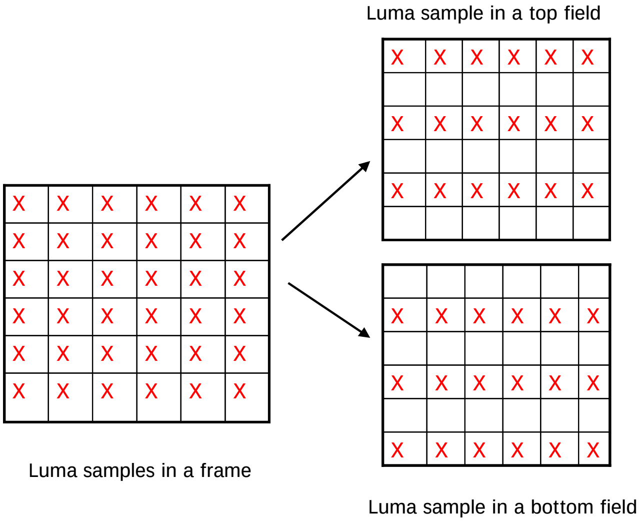


Fig. 3.9 The field packaging

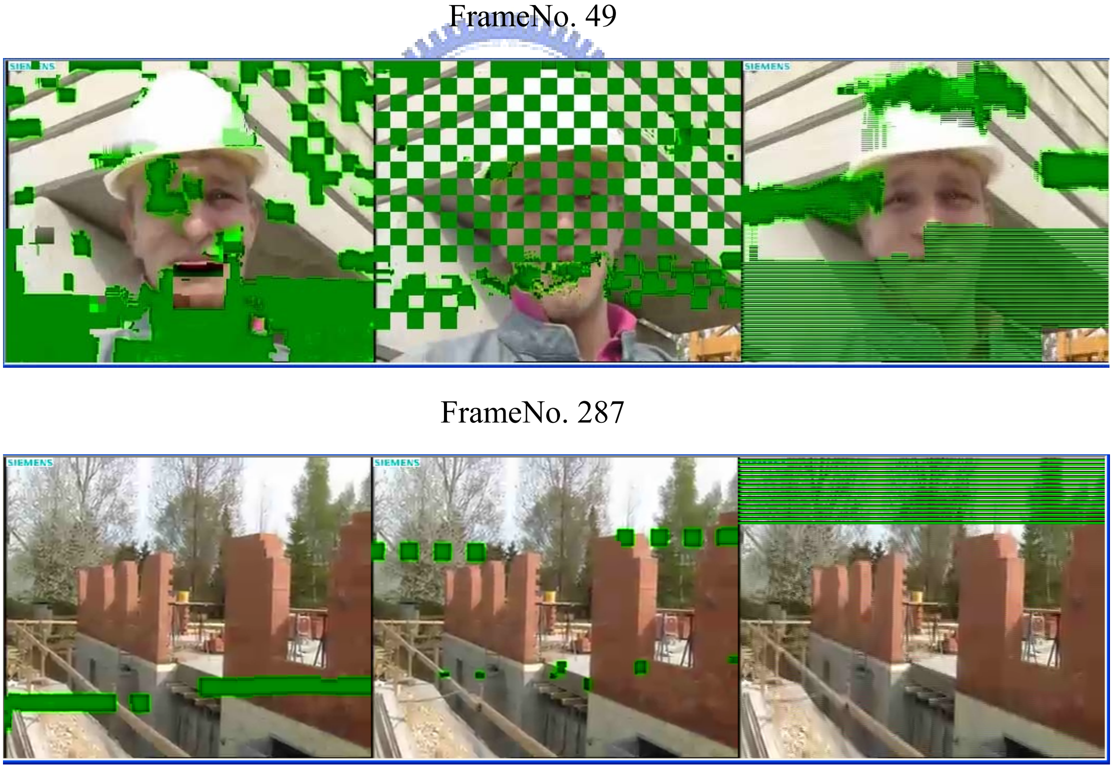
(Each block represents a pixel.)

Table 3.1 Coding rate records
Test sequences: Foreman, Stefan, Table, Container

	Frame coding	FMO2	FMO3	H.264 default field coding
Foreman	1	1.33	1.388	1.306
Stefan	1	1.32	1.29	1.08
Table	1	1.24	1.26	1.78
Container	1	1.32	1.421	3.49
Coding rate ratio	1	1.30	1.34	1.91

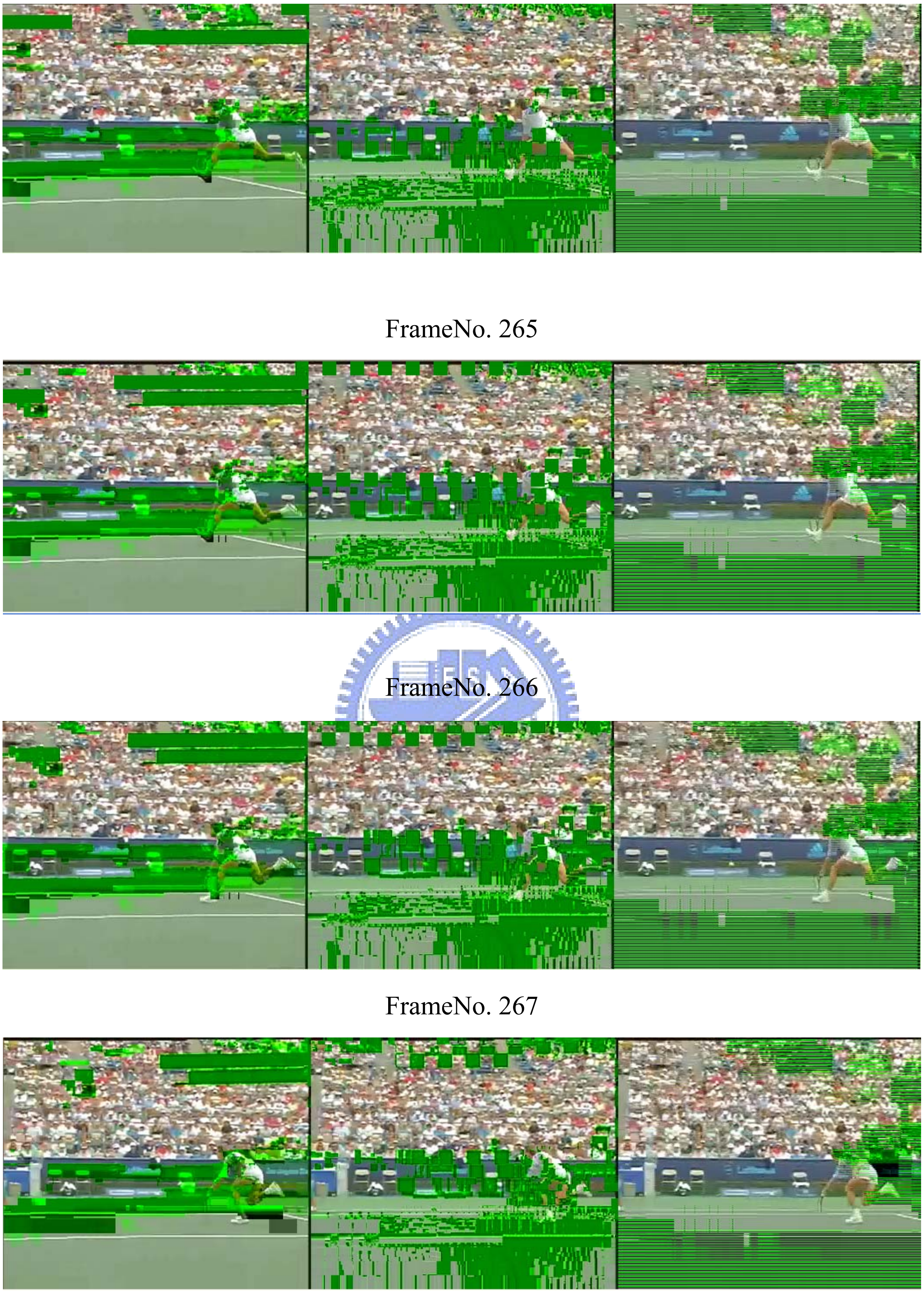
Experiment Results are shown in Fig. 3.10 and Fig. 3.11, including frame coding, FMO, and field coding.

FrameNo. 49



FrameNo. 287

Fig. 3.10 Experiment Results (Foreman)
From left to right: frame coding, FMO, and field coding



FrameNo. 264

FrameNo. 265

FrameNo. 266

FrameNo. 267

Fig. 3.11 Experiment Results (Stefan)
From left to right: frame coding, FMO, and field coding