Error Resilience Performance Evaluation of H.264 I-frame and JPWL for Wireless Image Transmission

Khalid Mohamed Alajel, Wei Xiang, and John Leis
Faculty of Engineering and Surveying
University of Southern Queensland
Toowoomba, QLD 4350, Australia
E-mail: {*khalid.alajel, wei.xiang, john.leis*}*@usq.edu.au*

Abstract— **The visual quality obtained in wireless transmission strongly depends on the characteristics of the wireless channel and on the error resilience of the source coding. The wireless extensions of the JPEG 2000 standard (JPWL) and H.264 are the latest international standards for still image and video compression, respectively. However, few results have been reported to compare the rate-distortion (R-D) performance of JPEG 2000 and H.264. Conversely, comparative studies of error resilience between JPWL and H.264 for wireless still image transmission have not been thoroughly investigated. In this paper, we analyse the error resilience of image coding based on JPWL and H.264 I-frame coding in Rayleigh fading channels. Comprehensive objective and perceptual results are presented in relation to the error resilience performance of these two standards under various conditions. Our simulation results reveal that H.264 is more robust to transmission errors than JPWL for wireless still image transmission.**

Index Terms—H.264, JPWL, error resilience, I-Frame coding, and wireless transmission.

I. INTRODUCTION

Transmission of multimedia data over wireless channels is becoming more pervasive. Multimedia signals are normally heavily compressed before transmission due to bandwidth constrains. Compressed data transmitted over wireless channels suffer from a number of degradations that are prevalent in wireless communications, caused the loss of synchronization between the decoder and encoder. This problem could be solved through retransmission. However, it results in increased delay, which is not acceptable for some real-time applications. An alternative approach is to use effective data protection to create compressed bitstreams resilient to transmission errors. As a result, error-resilient image and video data transmission becomes a crucial issue.

The wireless extension of the JPEG 2000 standard is commonly referred to as JPWL [1]. JPWL is the newest international standard for still image compression. Its goal is to allow for efficient transmission of JPEG 2000 coded images. The error resilience tools provided by the JPEG 2000 baseline can only detect the occurrence of errors, conceal erroneous data, and resynchronize the decoder. These tools fail to correct transmission errors and do not address the appearance of errors in the image header, although the header is the most important part of the codestream. Therefore, the error resilience tools provided by JPEG 2000 are not sufficient for wireless image transmission. To overcome the limitations of the JPEG 2000 baseline, JPWL provides additional tools for error protection and correction. The output from the JPWL encoder is a JPWL codestream which is robust for transmission over error-prone wireless channels.

The H.264/AVC advanced video coding is the latest international video coding standard developed by the JVT (Joint Video Team) consisting of experts from VCEG (Video Coding Expert Group) and MPEG (Moving Picture Expert Group) to provide better compression of video and error robustness over previous standards. H.264/AVC supports various applications such as video broadcasting and streaming over fixed and wireless networks. For particular applications, H.264/AVC defines a series of profiles and levels that place restrictions on the encoded bitstream. There are seven profiles for various application scenarios [2]. Noisy channels like wireless networks destroy the integrity of a received bitstream. The corrupted data degrades the subjective quality and propagates to the subsequent picture due to the use of predictive coding. Thus, H.264/AVC adopts some error resilience tools to reduce distortion resulting from errors and their propagation.

The JPWL as an extension of JPEG 2000 still image coding standard aims to allow for robust transmission of coded bitstreams over error-prone channels by providing some new error resilience tools. However, the success of the JPWL standard relies primarily on its improved error-resilient tools compared to previous image coding standards. Although several studies [3]–[7] have been focused on the coding efficiency of JPEG 2000 for still image coding, little is known about JPWL and especially about its error-resilient performance. On the other hand, H.264/AVC focuses on the coding of video sequences, but can also compress images by using the intra-coding mode. The error resilience performance analysis of JPWL and H.264/AVC I-frame, especially comparative studies between two standards have not been thoroughly conducted in the literature. Available literature is very limited in this regard. Most of the studies on H.264/AVC I-frame were intended to evaluate the rate-distortion (R-D) performance of H.264/AVC I-frame in comparison to other coding standards such as JPEG 2000 and MPEG-4 [8], [3], [5]. In those studies little was done regarding the error resilience tools in both standards. In [9] the authors investigated the performance of Motion-JPEG 2000 and MPEG-4 in error-prone transmission. The most important difference between those two standards is that all frames in Motion-JPEG 2000 are coded independently from each other (I-frame).Therefore, transmission errors in one frame do not propagate to subsequent frames. Error resilience tools in Motion-JPEG

2000 can only limit the effect of transmission errors but cannot correct them. In [10], error resilience tools in H.264/AVC and MPEG-4 are studied. The effect of instantaneous decoding refresh (IDR) slice, picture segmentation (PS), data partitioning (DP), and flexible macroblock ordering (FMO) have been tested. The results show that the usage of error resilience tools comes at the expense of increasing the output bitrate. More recently, the authors in [11] examine the error-resilient performance of the JPEG, JPEG 2000, and JPWL standards in wireless channels. Their results demonstrate that JPWL is more robust to transmission errors than JPEG and JPEG 2000. In this paper, we mainly focus upon evaluating the performance of H.264/AVC's error resilience tools and comparing against that of JPWL.

The remainder of this paper is organized as follows. Error resilience tools in H.264 and JPWL standards are briefly discussed in Section II. Section III presents the system configuration of the wireless image transmission system. Simulation results are presented in section IV. Finally, Section V concludes this paper.

II. ERROR RESILIENCE TOOLS IN JPWL AND H.264

In order to achieve high compression efficiency, compression algorithms aim to remove redundancy in the bitstream. In contrast, error resilience tools add extra information to the bitstream to limit the impact of errors. The purpose of error resilience tools in H.264/AVC and JPWL is to combat transmission errors through: detecting errors when they occur, concealing the erroneous data, desynchronizing the decoder, and correcting transmission errors. Compressed bitstreams are very sensitive to transmission errors, due to the use of variable-length coding (VLC) as the entropy coding scheme. The nature of VLC is the root cause of the phenomenon of error propagation. Even a single bit error could render the entire bitstream undecodable in the worst scenario. There are a number of error resilience tools adopted in the current image/video standards to make compressed bitstream more robust to channel errors. In what follows, we briefly review the error resilience tools provided by JPWL and H.264/AVC.

A. *Error Resilience Tools in JPWL*

The error resilience tools provided by JPEG 2000 baseline are unable to tackle channel errors occurred in the image header [12], [13]. Therefore, these tools are ineffective in wireless environments where the image header is often corrupted. The JPWL standard is designed to overcome this limitation. JPWL specifies a set of new error resilience tools and methods to prevent the coded bitstream against transmission errors, e.g., forward error correction (FEC), interleaving, and unequal error protection (UEP). These tools are informative, which means they are not compulsory under the JPWL standard. JPWL also defines a means of describing the sensitivity of the codestream to channel errors, and defines the locations of residual errors. In the presence of channel errors, error protection and correction such as cyclic redundancy check (CRC) and Read-Solomon (RS) codes were used to protect the main and tile part headers.

In JPWL, four new marker segments have been defined in the JPWL syntax namely, error protection capability (EPC), error protection block (EPB), error sensitivity descriptor (ESD), and residual error descriptor (RED). The EPC marker indicates whether the three other segments (ESD, EPB, and RED) are used in the codestream. EPC is a unique and its value is 0xFF97 in hexadecimal. Furthermore, it can also be used to signal the use of informative tools which have been registered with JPWL. EPB is used to protect the main and tile-part headers, which has a unique identifier of 0xFF96. The EPB marker segment is used to carry parameter information and parity data of the RS codes [14], which are the designated FEC codes in JPWL. The RS (160, 64) code is used to protect the first marker segment in the main header, RS (80, 25) code is used to protect the marker segment of a tile-part, and the RS (40, 13) is used to protect the other EPB marker segment for main and tile-part headers. The ESD marker segment with a unique identifier of 0xFF98 contains information about the sensitivity of codestream to errors. This information can be exploited by the decoder when applying UEP techniques where more powerful codes are used to protect the more sensitive portions of the codestream. However, the usage of error sensitivity in JPWL is not specified and this information is not essential to decode a codestream. Finally, the RED is a unique marker, whose value is 0xFF99. When the JPWL decoder fails to correct all errors in a codestream, RED signals the location of such errors. This information can be used by the JPWL decoder to better cope with errors [1].

B. *Error Resilience Tools in H.264/AVC*

H.264/AVC provides several error resilience tools that are mainly contained in the video coding layer (VCL), some of these tools are inherited from earlier video coding standards such as DP [15], [16]. Others are some new tools, i.e., FMO [17], and PS (slices) [18]. In this paper we will concentrate on error resilience tools which affect only still images since the purpose of the paper is to compare the error resilience tools of JPWL and H.264 for still image transmission.

1) Flexible Macroblock Ordering: flexible macroblock ordering (FMO) is one of the most interesting error resilience tools adopted in the H.264/AVC standard. FMO allows to partition macroblocks (MBs) in one frame into separate groups of MBs known as slice groups (SGs). This is unlike previous standards, in which the encoder is restricted to encode the MBs of a picture in the raster scan order. In the raster scan, the encoder starts at the upper left corner of a picture and then processes the MBs raw by raw until the bottom right corner. Using FMO, MBs are no longer assigned to slices in raster scan order. Instead, every MB is assigned freely to a specific SG using a macroblock allocation map (MBAmap). In H.264/AVC, SG introduces a new layer between each picture and its slices, which means that the pictures are not divided into slices but into slice groups instead [17]. At the decoder side, the decoder should know which macroblock is assigned to which slice group by transmitting the MBAmap together with the coded macroblocks. The objective of FMO is to scatter possible errors to the whole frame to avoid error accumulation in a limited region [16]. This is because it is hard to conceal concentrated errors in a small region compared to scattered ones. H.264 specifies seven different types of FMO labeled types 0 to 6 [18]. The first six types are patterns, which can be exploited when storing and transmitting the MBAmap. The last one is the most general type used, when the map cannot be described by the first six ones and should be transported completely.

2) Data Partitioning: VCL is clearly distinct from the network abstraction layer (NAL) in the H.264/AVC standard [18]. Normally, each slice is put into separate network abstraction layer unit (NALU), which consists of a one-byte header followed by payload data. Data partitioning (DP) in H.264/AVC allows the partition of a normal slice into three parts (data partitioning A, B, and C), which are then encapsulated into separate NAL [19]. Data partition is achieved by separating the coded slice data (macroblock, header information, motion, and texture information) into separate sections. The idea of data partitioning is that when one partition is lost, is still able to use information from the correctly received partitions. Data partition A contains the slice header, macroblock types, quantization parameters, prediction modes, and motion vectors. Thus, the loss of partition A means the data of other partitions becomes useless. Partition B contains residual information of intra-coded macroblocks, so the loss of partition B will only affect the recovery of successive frames. Data partition C contains residual information. This dependency can be avoided by restricting the encoder to use only residual data from other intra-coded MB for inter-coded macroblocks which is less important compared to other data contained in each slice. However, it is the biggest partition of the coded slice due to the large number of frames coded as P-frames. In the case of intra prediction, pixels from surrounding MBs are used to predict the current MB. This means a dependency between partitions B and C. Partition B is independent from partition C. On the other hand, no option is available to make partition C independent from partition B. Partition A is completely independent of partitions B and C.

3) Picture segmentation (Slicing): A picture may be divided into one or more slices, where a slice has a header and data partition. Each slice consists of a given number of MBs and a data partition contains one MB or a sequence of MBs. A picture consists of one up to seven SGs, which are independently decodable and thus important to prevent propagation of errors. In picture segmentation (PS), a slice may be encoded as I, predictive (P), or bidirectional (B) slices depending on the nature of MBs belonging to the slices. For I slices, all MBs are coded using intra prediction. For P and B slices, MBs can be coded using either intra or inter prediction. Slices are used as error resilience tools in the H.264/AVC standard to prevent propagation of errors. However, error resilience tools introduce some overhead to the compressed bitstream and reduce coding efficiency, but in error-prone environments the quality of received data can be greatly improved.

III. SYSTEM CONFIGURATION

This section is focused on investigating the error resilience performance of H.264 I-frame and JPWL through simulations. Our simulation program is based upon the open-JPEG library for JPWL [20] and JM reference software for H.264 [21]. The encoded data are then transmitted over Rayleigh fading channel to evaluate the performance of error resilience tools provided by the standards. The overall system is depicted in Fig. 1.

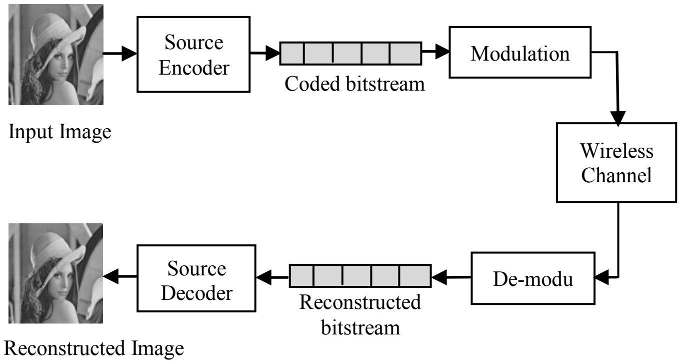


Fig. 1. Image transmission over wireless system.

For JPWL, the open-JPEG library is used as an encoder/decoder for JPWL which is open-source JPEG 2000 software written in C and developed by the communications and remote sensing Lab, Louvain-la-Neuve, Belgium [20]. Cyclic redundancy checks (CRCs) are used as an error detection tool to detect the occurrence of errors in coded images. The two CRC options available, i.e., CRC-16 and CRC-32 codes are used to ensure that packets with errors are detected and thus not decoded. On the other hand, for correcting the detected errors, Reed-Solomon codes are the inherent error correction code supported by JPWL to correct transmission errors. The supported RS codes include RS (37,32), RS (38,32) , RS (40,32) , RS (43,32) , RS (45,32) , RS (48,32) , RS (51,32) , RS (53,32) , RS (56,32) RS (64,32) , RS (75,32) , RS (80,32) , RS (96,32) , RS (112,32), and RS(128,32). Three different methods will be used to test the performance of JPWL in regard to wireless image transmission .i.e., no protection, CRC protection, and Reed-Solomon protection.

The encoder and decoder of H.264 have been implemented in software based on the standard JVT codec software version 13.1[21]. It has been evaluated in different coding scenarios using different error resilience source coding tools. We have arranged for a series of tests to evaluate the robustness of compressed bitstreams against transmission errors. The resilience of a bitstream is first presented without error resilience tools, and then the performance of DP, FMO, and PS on the transmitted bitstream is presented.

To simulate the channel errors, Dent's model [22] has been used to model the Rayleigh fading channel with additive white Gaussian noise. The Rayleigh fading is a good model of wireless communication when there are many objects in the environment that scatter the transmitted signal before it arrives at the receiver. The Rayleigh fading channel is modeled using a modification of Jakes model [23] proposed by Dent *et. al.* [22]. The objective of Dent's model is to remove the cross correlation between waveforms in the Jakes's model and can be mathematically expressed as

$$S(t)=\sqrt{\frac{2}{N_0}}\sum_{n=1}^{N_0}[\cos\beta_n+i\,\sin\beta_n]\,\cos(\omega_n t+\theta_n) \qquad (1)$$

where $N_0 = N/4$ is the number of complex oscillators and N the number of arriving rays. β_n are the phases and given as

$\pi n / N_0$ and θ_n are the initial phases which normally set to zero.

As an objective performance measure for quality comparison of two codecs, PSNR of the luminance component was chosen to represent the visual quality. The PSNR is defined as

$$\mathrm{PSNR}=10\log_{10}\left(\frac{255^2}{MSE}\right)\ \text{(dB)} \tag{2}$$

where the mean square error (MSE) has been computed on the Y (luminance) component only.

IV. EXPERIMENTAL RESULTS

This section describes the experimental procedure and the simulation environment for wireless imaging using JPWL and H.264/AVC I-frame. The performance of the above error resilience tools for both standards in Rayleigh fading channels based upon Dent's model is evaluated and results are presented in this section.

A) Parameter Settings

The first codec is JVT implementation of the H.264/AVC standard JM version 13.1. Details of the codec parameters are given in Table I. The second is the open-JPEG library of JPWL. The quadrature amplitude modulation (QAM) and AWGN channel model were written in MATLAB. Two test images (monochrome), namely Lena and Boat in bmp format of resolution 256 x 256 pixels are used to test the proposed system.

Results of 60 simulations, performed with different channel seeds, are averaged in order to obtain more reliable results, the average PSNR value is given by

$$\mathrm{Average\_PSNR}=\frac{1}{60}\sum_{N=1}^{60}\mathrm{PSNR}(N) \tag{3}$$

TABLE I
H.264/AVC CODEC PARAMETERS

Parameter	value
Intra period	enabled
Entropy coding	CAVLC
Profile ID	Extended profile
Number of P and B frames	disabled
Rate distortion optimization	enabled

B) Results and discussions

In this section, through various simulations we intend to demonstrate the coding efficiency and error resiliency properties of H.264 and JPWL.

1) Rate distortion efficiency: To evaluate the performance of H.264/AVC I-frame and JPWL for coding still images, the R-D curves of the luminance components are computed. The H.264 encoder was run with the parameters set out in Table I and with a fixed quantization step size (QP). Then the H.264/AVC decoder calculated the bitrate and PSNR date. The target bitrate for JPWL encoder was computed using the bitrate from H.264/AVC decoder.

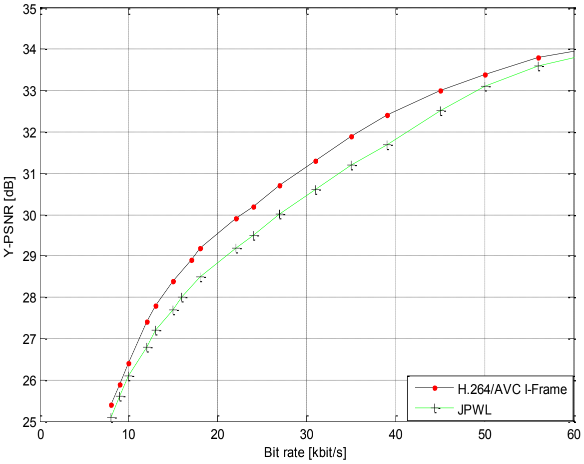


Fig. 2. Rate distortion curve for Lena image.

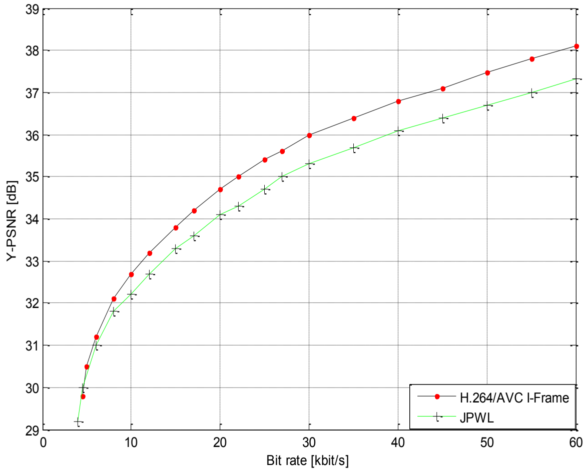


Fig. 3. Rate distortion curve for Boat image.

Fig. 2 depicts the R-D curves of our coding experiments for Lena test image. The PSNR of H.264 I-frame is higher than JPWL for about 0.1 dB to 0.8 dB at all of tested bitrates. The PSNR gains reduce while the bitrates decreases. However, at very low bitrates, JPWL outperforming H.264 in terms of PSNR and recover some of the distance at very high bitrates. Similar results for Boat image are shown in Fig. 3.

Subjective results presented in Fig. 4 show Lena images comparing the coding efficiency of H.264 Intra coding (a) and JPWL (b) at 22 kbits per image, the obtained results show superiority of H.264. PSNR values for Lena coded by JVT JM codec (29.4 dB) is higher than PSNR for the same image coded by JPWL codec (28.8 dB).

2) Evaluation of error resilience tools: Table II presents PSNR results for the Lena image without protection, CRC-16, and CRC-32 codes in JPWL under different values of SNR. As results show, CRC codes presented similar values to the case of no protection. This can be attributed to the fact that CRC codes are only used to detect the occurrence of errors but they do not provide any error correction capabilities.

(a) H.264 I-frame, 29.4 dB (b) JPWL, 28.8 dB

Fig. 4. Subjective results of Lena image comparing H.264/AVC Intra coding and JPWL at 22 kbits per image.

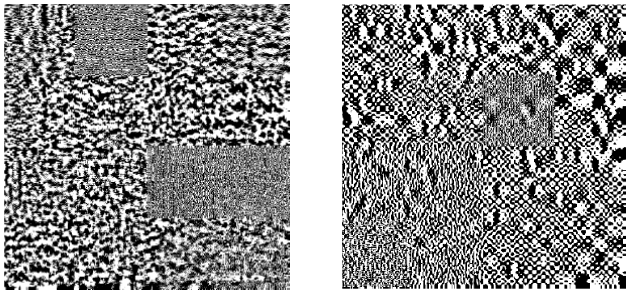
RS codes are used as the error correction codes to protect the images from channel errors. The performance of RS codes improves the quality of reconstructed images by increasing the redundancy in RS code. For example, RS (37, 32) improves the PSNR values by about 0.01 dB whereas RS (64, 32) increase the PSNR by 5.49 dB compare to no protection option at SNR= 9 dB. Fig. 5 demonstrates subjective results of Boat image using: no protection, CRC-32, RS (37, 32), and RS (64, 32) at SNR= 21 dB.

Regarding H.264/AVC we have examined the effect of DP, FMO, and PS error resilience tools. Other error resilience tools supported by H.264/AVC like IDR have not been tested because they do not effect on still images. The PSNR values and output bitrate have been calculated at the encoder output as detailed in Table III. From these results it is clear that the use of every error resilience tool increases the output bitrate. For example, the use of FMO increase the output bitrate by about 2 kbit (QP = 20). Also the usage of all error resilience tools together increasing the bit rate significantly. The increased bitrate needed to transmit image is a trade-off for better image quality.

As can be seen from Table IV, it is very clear that the performance of H.264 is significantly better than of JPWL in terms of error resilience tools. The PSNR error performance of DP, FMO, and PS in H.264 are also evaluated and numerical results are shown for Lena image. As we can see, all introduced error resilience tools improve the image quality in comparison with no error resilience components.

TABLE II
COMPARSION OF AVERAGE PSNR FOR LENA IMAGE USING NO PROTECTION AND CRC CODES IN JPWL

CHANNEL SNR (dB)	AVERAGE PSNR (dB)		
	No Protection	CRC-16	CRC-32
9	5.79	5.85	5.79
12	5.86	5.83	5.69
15	5.72	5.77	5.74
18	5.78	5.57	5.78
21	6.83	5.92	6.93
24	10.47	9.86	9.12
27	13.26	15.61	13.10
30	20.93	17.37	18.87
33	25.41	23.99	24.42
36	28.40	30.27	30.05
39	32.49	31.56	31.48



(a) No Protection (b) CRC-32

(c) RS (37, 32) (d) RS (64, 32)

Fig. 5. Subjective results of Boat image using: no protection, CRC-32, RS (37, 32), and RS (64, 32) at SNR =21 dB.

TABLE III
OUTPUT BITRATE VALUES AT H.264/AVC ENCODER FOR LENA IMAGE

QP	Output bitrate [Kbit]				
	No Error Resilience	Error Resilience Tools			
		DP	FMO	PS	ALL
10	77.600	77.603	80.296	78.216	80.376
20	37.080	37.083	39.256	37.712	39.336
30	16.936	16.936	18.224	17.232	18.304
40	7.176	7.178	8.000	7.416	8.080
50	3.376	3.376	3.880	3.544	3.960

TABLE IV
COMPARSION OF AVERAGE PSNR FOR LENA IMAGE USING NO PROTECTION, DP, FMO, AND PS IN H.264

CHANNEL SNR (dB)	AVERAGE PSNR (dB)			
	No Protection	DP	FMO	PS (10 slices)
9	6.51	7.36	6.91	6.67
12	6.63	7.43	7.01	6.86
15	7.13	8.26	7.59	7.26
18	8.19	11.66	10.24	10.13
21	12.01	16.59	14.28	14.38
24	12.49	19.24	16.81	17.36
27	18.48	25.01	22.14	22.42
30	25.01	32.76	29.37	29.65
33	32.14	38.08	34.43	35.26
36	41.61	42.24	40.37	40.59
39	44.37	45.13	44.26	43.37

Fig. 6 compares the performance of DP error resilience for Lena image coded bitstream for a range of SNR values. It can be observed that in terms average PSNR, enabling the DP which is simple with negligible overhead as shown in Table III and introduce small quality degradation has significantly improved the resilience of coded image against channel errors. DP offers about 8.5 dB gain against no error resilience.

From Table III, it is obvious that FMO introduced more overhead and hence, the image quality is degraded compared to DP. But in comparison with no error resilience, FMO offers about 3.5 dB gain. The H.264 standard can assign MBs to slice groups. The simplest FMO has two slice groups, but more groups are possible in the standard. Fig. 7 shows how FMO can improve the resilience of Lena coded image against channel errors where two slice groups are used.

Fig. 8 demonstrates the average PSNR values for average of SNR values for Lena image. Three different values of slices have been inserted to the encoded image before transmission. It is obvious from the obtained results that the output quality in error-prone situations has been improved. In contrast, this will increase the overhead data added to coded bitstream. Therefore, in the case of error-free transmission or near to error-free, the output quality will be degraded.

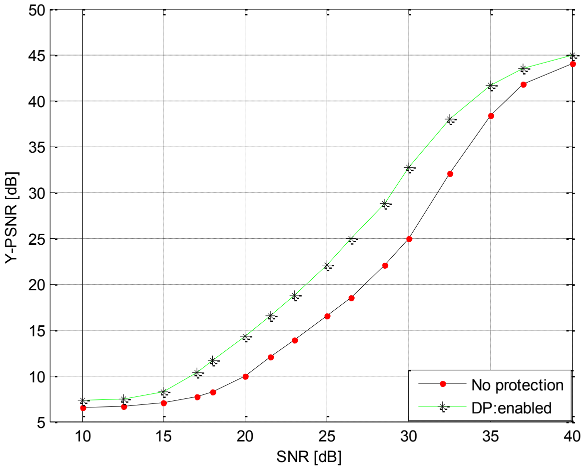


Fig. 6. PSNR vs. SNR for Lena image when DP is enabled and disabled.

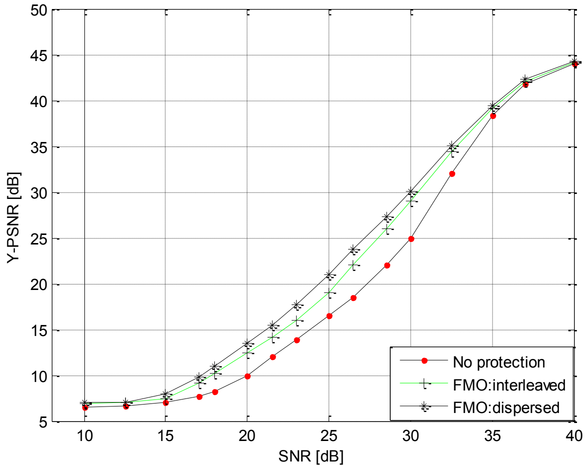


Fig. 7. PSNR vs. SNR for Lena image when FMO is enabled and disabled.

As can be observed from the figure, at low SNR, slicing significantly improves the performance. The highest average PSNR has been achieved when the channel SNR is between 15 dB and 32 dB with the slice number is equal to 15 slices. At high SNR (near error-free) adding more slices come with more overhead and reduced the quality. It can be seen that after SNR = 35 dB the PSNR values are reduced by increasing the number of slices.

Subjective results in Fig. 9 presents the objective results mentioned in the previous paragraphs for the Boat image. Fig. 9 (a) illustrates the transmitted image over the same channel with SNR=21 dB without error resilience tools, whilst Fig. 9 (b), (c), and (d) show the effect of DP, FMO, and PS respectively and how the subjective quality of the received image is improved.

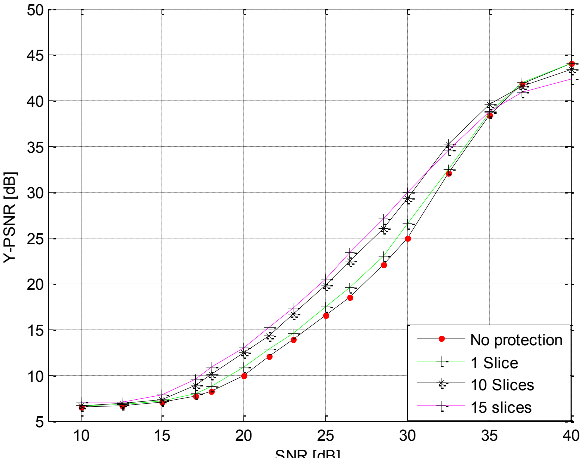


Fig. 8. PSNR vs. SNR for Lena image with different slice mode.

(a) No protection

(b) DP

(c) FMO

(d) PS: 1 Slice

Fig. 9. Subjective results of Boat image using: no protection, DP, FMO, and PS at SNR =21 dB.

V. CONCLUSIONS

This paper focuses on evaluating the coding efficiency and error resilience of JPWL and H.264. Comprehensive objective and subjective results are presented to examine the performance of error resilience tools in these two standards for wireless image transmission over Rayleigh fading channels. Demonstrated results show the superiority of H.264 in terms of coding efficiency compared with JPWL. Moreover, the results obtained indicate that H.264 is much more robust to transmission errors than JPWL. The error resilience tools can significantly improve the quality of reconstructed images. Although there is some overhead introduced to the coded bitstream to cater for error resilience, a tradeoff is made for better quality. We conclude that H.264 is more suitable for still image transmission over error-prone channels and in error-prone environments than JPWL.

REFERENCES

[1] F. Dufaux, G. Baruffa, F. Frescura, and D. Nicholson, “JPWL - an extension of JPEG 2000 for wireless imaging,” in Proc. IEEE International Symposium on Circuits and Systems (ISCAS’06), Kos, Greece, May 2006, pp. 3870–3873.

[2] A. Tamhankar, and K.R. Rao, “An overview of H.264/MPEG-4 Part 10,” in Proc. 4th EURASIP Conference focused on Video/Image Processing and Multimedia Communications, July 2003, vol.1, pp. 1- 51.

[3] O. Mourad, D. Frederic, and E. Touradj, “On comparing JPEG 2000 and Intraframe AVC,” in Proc. SPIE of Digital Image Processing, Aug 2006.

[4] T. Trac, L. Lijie, and T. Pankaj, “Performance comparison of leading image codecs: H.264/AVC Intra, JPEG 2000, and Microsoft HD Photo,” In Proc. SPIE Video and Image Technologies, Vol. 6696, Sep 2007.

[5] C. Snag, B. Zoran, M. Dragorad, L. Jungsik, and H. Jae, “Image quality evaluation: JPEG 2000 versus intra-only H.264/AVC high profile” The scientific journal, Facta Universities. Elec Energ., vol. 20, no. 1, pp.71-83, April 2007.

[6] S. Boxin, L. Lin, and X. Chao, “Comparison between JPEG 2000 and H.264 for digital cinema,” In Proc. IEEE International Conference on Multimedia and Expo (ICME 08), April 2008, pp. 725-728.

[7] J. Runhai, “Performance Comparison Between AVC I-Frame Coding and JPEG 2000,” *In Proc. Second International Conference on Computer Modeling and Simulation, (ICCMS '10),* Jan 2010, pp. 315-318.

[8] D. Marpe, V. George, H. L. Cycon, and K. U. Barthel, “Performance evaluation of motion-JPEG 2000 in comparison with H.264/AVC operated in pure intra coding mode,” *In Proc. SPIE Wavelet Applications in Industrial Processing,* Rhode Island, USA, October 2003.

[9] D. Frederic, and E. Touradj, “Error-resilient video coding performance analysis of Motion-JPEG200 and MPEG-4,” *In Proc. SPIE Visual Communications and Image Processing,* San Jose. CA, June 2004.

[10] J. Mochnac and S. Marchevsky, "Error resilience tools in the MPEG-4 and H.264 video coding standards," *in Proc. 18th International Conference on Radioelektronika,* April 2008, pp. 1-4.

[11] X. Wei, A. Clemence, J. Leis, and W. Yafeng, “Error resilience analysis of wireless image transmission using JPEG, JPEG 2000 and JPWL,” *in Proc. 7th International Conference on Information, Communications and Signal Processing (ICICS 2009)*, Macua, Dec 2009, pp. 1-6.

[12] I. Moccagatta, S. Soudagar, J. Liang, and H. Chen, “Error-resilient coding in JPEG-2000 and MPEG-4,” IEEE J. Select. Areas Commun., vol. 18, no. 6, pp. 899–914, June 2000.

[13] A. Skodras, C. Christopoulos, and T. Ebrahimi, “The JPEG 2000 still image compression standard,” *IEEE Magazine. Signal Processing. , vol. 18, no. 5, pp. 36-58,* Sep 2001.

[14] S. Lin and D. J. Costello, Jr., Error Control Coding: Fundamentals and Applications, 2nd ed., Prentice-Hall, Upper Saddle River, NJ, 2004.

[15] G. Yi, C. Ying, W. Ye-Kui, L. Houqiang, M.M. Hannuksela, and M. Gabbouj, “Error-resilient Coding and Error Concealment in Scalable Video Coding,” *IEEE Trans. Circuits Syst. Video Technol., vol. 19, no. 6, pp.* 781-795, June. 2009.

[16] S. Kumar, L. Xu, M. K. Mandal and S. Panchanathan, "Error Resiliency Schemes in H.264/AVC Video Coding Standard", *Elsevier Journal of Visual Communication and Image Representation (Special issue on Emerging H.264/AVC Video Coding Standard), Vol. 17(2), pp. 425-450,* April 2006.

[17] P. Lambert, W. De Neve, Y. Dhondt, and R. Van de Walle, “Flexible macroblock ordering in H.264/AVC,” *Elsevier Journal of Visual Communication and Image Representation, vol. 17, no. 2, pp. 358-375,* April 2006.

[18] S.-k. Kwon, A. Tamhankar, and K.R. Rao, “Overview of H.264/MPEG-4 part 10,” *Elsevier Journal of Visual Communication and Image Representation*, vol. 17, no. 2, pp. 186-216, April 2006.

[19] S. Wenger, “H.264/AVC over IP,” *IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 7, pp. 645-656.* July 2003.

[20] The OpenJPEG Team, The Communications and Remote Sensing Lab.Louvain-la-Neuve, Belgium. [Online]. Available: http://www.openjpeg.org/index.php?menu=main

[21] JM 13.1 JVT Reference Software. [Online]. Avaliable: http://iphome.hhi.de/suehring/tml/download/old_jm/

[22] P. Dent, G. Bottomley, and T. Croft, “Jakes fading model revisited,” *Electron. Lett., vol. 29, no. 13, pp. 1162–1163*, June 1993.

[23] W. C. Jakes, Microwave Mobile Communications, Wiley, New York, 1974.