MOTION JPEG2000 FOR WIRELESS APPLICATIONS

Frédéric Dufaux and Touradj Ebrahimi

Signal Processing Laboratory - Swiss Federal Institute of Technology (EPFL)
CH-1015 Lausanne, Switzerland

Emitall SA
Rue du Théâtre 5, CH-1820 Montreux, Switzerland

ABSTRACT

In this paper, we analyze the performance of Motion JPEG2000 for wireless applications. This new standard is based on intra-frame wavelet coding. Motion JPEG2000 is offering a number of very compelling advantages when compared to state-of-the-art MPEG-4 video coding. Wavelet coding achieves very high coding efficiency. Furthermore, because frames are intra coded, it is very error-resilient. Finally, Motion JPEG2000 requires low complexity, supports very efficient scalability, and introduces minimal coding delay. In this paper, we propose an analysis of the performance of Motion JPEG2000 and a comparison with MPEG-4 in terms of coding efficiency, error resilience and complexity. We present experimental results which show that Motion JPEG2000 outperforms MPEG-4 for wireless applications.

1. INTRODUCTION

The recent Motion JPEG2000 standard [1] is offering a number of very compelling features. In this paper, we study its performance for the transmission of video over low bit rate error-prone wireless channels. In particular, we carry out a comparison with the well-known MPEG-4 standard.

Motion JPEG2000 is an extension of JPEG2000 for the coding of video sequences. It is based on an intra-frame wavelet transform. The reader is referred to [2] for a thorough presentation of JPEG2000. As it relies on intra-frame coding, Motion JPEG2000 has a lower coding efficiency at the benefit of a lower complexity. In addition, because frames are coded independently, preventing the propagation of errors across consecutive frames, Motion JPEG2000 is more error-resilient. Finally, Motion JPEG2000 provides with additional features such as various forms of scalability (e.g. resolution or quality) and very precise rate control which results in minimal coding delay.

Conversely, MPEG-4 is based on a motion compensated block-based Discrete Cosine Transform (DCT). A detailed description of MPEG-4 can be found in [3]. By using motion compensation to reduce inter-frame redundancy, MPEG-4 achieves a higher coding efficiency but requires a higher computational complexity. Besides, the temporal prediction loop results in dependencies between coded frames. Therefore, the effects of transmission errors propagate across consecutive frames and MPEG-4 is characterized by lower error resilience.

In order to more efficiently support wireless applications, a number of tools have been included in both Motion JPEG2000 and MPEG-4 to improve resilience to transmission errors. Basically, these tools detect where errors occur, conceal the erroneous data, and resynchronize the decoder. This is achieved by mechanisms such as resynchronization markers, data partitioning and Reversible Variable Length Codes (RVLC).

A study of the performance of JPEG2000 for still image coding has been presented in [4]. An evaluation of the error resilience performance of JPEG2000 and MPEG-4 for still image coding has been presented in [5], whereas the MPEG-4 error resilience tools have been addressed in [6]. The problem of video quality evaluation for mobile applications has been discussed in [7], along with the presentation of subjective test results for Motion JPEG2000 and MPEG-4. In this paper, we propose a thorough comparison of the performance of Motion JPEG2000 and MPEG-4 in the framework of video transmission over low bit rate error-prone wireless channels.

This paper is structured as follow. In Sec. 2, we briefly review Motion JPEG2000 and MPEG-4 error resilience tools. In Sec. 3, we introduce the simulation set-up used in our tests. Experimental results are presented in Sec. 4. Finally, we draw some conclusions in Sec. 5.

2. ERROR RESILIENT CODING

Compression and error resilience are two contradictory requirements. Obviously, the highest the compression ratio, the more important the resulting bits are. Hence, a highly compressed bit stream is generally more sensitive to transmission errors.

In order to alleviate this problem, error resilient coding tools have been proposed and adopted in the Motion JPEG2000 and MPEG-4 standards. These tools limit the impact of errors by detecting their occurrence, concealing the erroneous data, and re-synchronizing the decoder.

As most coding schemes, Motion JPEG2000 and MPEG-4 rely on Variable Length Codes (VLC). VLC are particularly sensitive to transmission errors. Indeed, transmission errors cause the loss of all the data between the occurrence of the error and the next position where the decoder can resynchronize. Therefore, in order to minimize the amount of data lost, resynchronization markers have to be frequently inserted in the bit stream. These markers are special codes which can be unequivocally recognized by the decoder.

If the decoding of a part of the bit stream depends on the correct decoding of previous parts of the bit stream, the effect of transmission errors will spread. More precisely, all parts of the bit stream which depend on the corrupted data will also be lost. Therefore, it is desirable to have coded units which can be independently decoded. This is commonly referred to as data partitioning.

2.1. Motion JPEG2000 error resilient tools

We now briefly review the error resilient tools in Motion JPEG2000. A more detailed description is given in [5].

The first and foremost advantage of Motion JPEG2000 is that it relies on intra-frame coding. As each frame is coded independently of the other ones, transmission errors in one frame do not propagate to subsequent frames.

Within each frame, the code stream is composed of packets. Each packet corresponds to a quality layer, a resolution, a component and a precinct. These packets constitute independently coded units. Therefore, this data partitioning limits the spread of transmission errors to a great extent. Furthermore, resynchronization markers can be inserted in front of every packet.

The quantized wavelet coefficients are grouped into code-block and coded using an MQ arithmetic coder. Each code-block is independently coded. To improve its error resilience, the arithmetic coder can be terminated and the contexts can be reset after each coding pass. Furthermore, it is also possible to encode a segment marker at the end of each coding pass. If the segment marker is not correctly decoded at the decoder side, an error is flagged in the preceding pass.

While the above tools detect where errors occur, conceal the erroneous data, and resynchronize the decoder, they do not correct transmission errors. Furthermore, these tools do not apply to the image header which is the most important part of the code stream. Per consequent, JPEG has started a new work item, JPWL, addressing these issues [8].

2.2. MPEG-4 error resilient tools

MPEG-4 has also a number of error resilient tools that we will now briefly discuss. A more thorough description can be found in [5][6].

MPEG-4 has a mechanism for resynchronization markers. The code stream is divided into video packets consisting of an integral number of macro-blocks, and the resynchronization markers can be inserted in front of these packets. In a preferred operating mode, resynchronization markers are placed periodically.

In MPEG-4, the code stream syntax typically combines together the DCT coefficients, the motion vectors, and the information about macro-blocks coding mode. While very efficient in terms of coding, this syntax is very weak in terms of error resilience. Indeed, when an error occurs, all the data is likely to be lost. To circumvent this problem, MPEG-4 has defined an alternative syntax based on the principle of data partitioning. In this case, all the motion vectors are grouped together on the one hand and all the DCT coefficients are grouped together on the other hand. The code stream is then composed of all the regrouped motion data, followed by all the regrouped DCT data, the two parts being separated by a marker. In this way, if the motion data is corrupted, all DCT data can still be correctly decoded. Conversely, if the DCT data has errors, all the motion data can still be recovered.

Another option of MPEG-4 is to use Reversible Variable Length Codes (RVLC). In RVLC, each codeword can be identically decoded in the forward and backward directions. In the presence of an error, the decoder search for the next resynchronization marker and then starts decoding the code stream in the backward direction. In this way, more data can be recovered compared to the case of standard VLC.

Finally, MPEG-4 has a mechanism, referred to as Header Extension Code (HEC), to repeat the header information. Indeed, the header contains critical information such as the spatial resolution of a frame, the picture type and the time stamp. Therefore, its loss results in the entire frame being dropped.

3. SIMULATION SET-UP

In order to simulate the transmission of video sequences over a WCDMA wireless channel and to analysis quantitatively the above considerations, we used the setup illustrated in Figure 1. Source encoding is followed by H.223 for multiplexing and packetization of the bit stream. The occurrence of transmission errors is simulated using bit error patterns representative of WCDMA [9].

Due to the random nature of transmission errors, we run a large number of trials which consist in applying random circular shifts to the same bit error pattern using different seeds. Final results are obtained by averaging over all the trials.

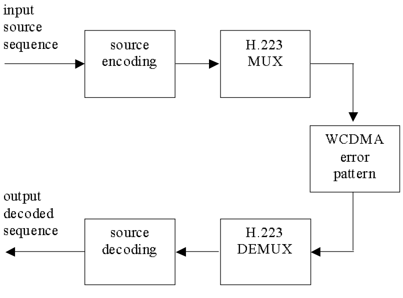

Figure 1: Simulation environment.

4. EXPERIMENTAL RESULTS

We ran simulations on 9 video sequences which cover a wide range of content. In particular, the sequences have been selected to encompass slow, moderate and fast object and camera motion. Hence, the scenes span a wide range of coding complexity. The sequences are in QCIF format and are encoded at 128 kb/s. We used the Kakadu software [10] for Motion JPEG2000 and the MoMuSys reference software [11] for MPEG-4.

4.1. PSNR results

Performance results are strongly content-dependent. In the case without transmission errors, MPEG-4 typically outperforms Motion JPEG2000 on sequences with low to moderate motion, whereas Motion JPEG2000 has comparable or even better performances than MPEG-4 on sequences with fast motion. Figure 2 show PSNR results without transmission errors for *Balloons*, a typical sequence with moderate motion.

In the presence of transmission errors, Motion JPEG2000 is gaining the upper hand over MPEG-4. Figure 3 show PSNR results in the presence of transmission errors for the same *Balloons* sequence. The Bit Error Rate (BER) is 1e-4 and the results have been averaged over 100 trials for each sequence.

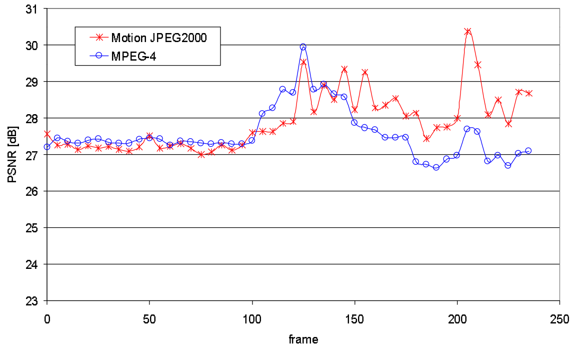

Figure 2: Motion JPEG2000 versus MPEG-4 without transmission errors.

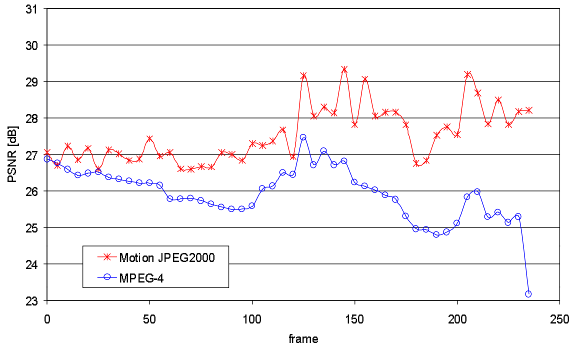

Figure 3: Motion JPEG2000 versus MPEG-4 with transmission errors (BER=1e-4).

PSNR results on all test sequences are summarized in Table 1. We can observe that in the case without transmission errors, MPEG-4 achieves on average a gain of 1.26 dB over Motion JPEG2000. In the presence of transmission errors, PSNR for Motion JPEG2000 drops by less than 0.5 dB, showing its strong error resilience. In the same conditions, the PSNR for MPEG-4 falls by almost 3 dB. Consequently, Motion JPEG2000 now outperforms MPEG-4 by an average gain of 1.22 dB.

	MJ2K	MP4	gain
Balloons	27.90	27.56	0.34
New York	35.64	37.28	-1.64
Mobile	23.11	26.49	-3.38
Animals	34.28	37.71	-3.43
Letters	27.32	26.18	1.14
Waterfall	29.93	32.86	-2.93
Football	32.99	31.37	1.62
Suzie	39.01	39.71	-0.70
Tempest	27.08	29.45	-2.37
average	30.81	32.07	-1.26

Table 1: PSNR without transmission errors.

sequence	MJ2K	MP4	Gain
Balloons	27.54	25.92	1.62
New York	35.11	31.83	3.28
Mobile	22.87	24.54	-1.67
Animals	33.69	32.45	1.24
Letters	26.93	24.92	2.01
Waterfall	29.65	31.15	-1.5
Football	32.51	29.46	3.05
Suzie	38.28	34.69	3.59
Tempest	26.72	27.32	-0.6
average	30.37	29.14	1.22

Table 2: PSNR with transmission errors (BER=1e-4).

4.2. Perceptual results

The above PSNR figures are now complemented by the use of perceptual metrics as well as visual inspection of the decoded sequences. A first observation is that Motion JPEG2000 and MPEG-4 results in very different artifacts. This is due to the fact that the underlying technology is very different: intra-frame wavelet for Motion JPEG2000, and motion compensated DCT for MPEG-4.

In Figure 4, we present results obtained with Genista's *Video PQoS* software [12]. In particular, we show perceptual metrics results for three types of artifacts: blockiness, blur and noise. The results are for the sequence *Balloons*, in the case with transmission errors (BER=1e-4). In the three graphs, the values along the vertical axis are in arbitrary units. A larger value means that the detection of the artifact is stronger. We can clearly see that MPEG-4 exhibits more block artifacts than Motion JPEG2000. This is highly expected, as block artifacts are a trademark of DCT whereas they are absent in wavelet-based coding. We can also see that Motion JPEG2000 tends to introduce slightly more blur than MPEG-4. This is due to the high compression of the high frequency coefficients in Motion JPEG2000. Finally, MPEG-4 introduces more noise in the coded sequence.

Figure 5 shows sample decoded frames from the sequences *Balloons* and *Suzie*, in the case with transmission errors (BER=1e-4). As can be observed, transmission errors tend to introduce some hazy waves in Motion JPEG2000. Conversely, they produce blocks of strange color or texture in MPEG-4, which are visually more annoying. Furthermore, in Motion JPEG2000, only the specific frame which has been hit with transmission errors exhibit distortions. In the case of MPEG-4, the distortions persist for several frames.

Extensive subjective test results for Motion JPEG2000 and MPEG-4 have also been presented in [7].

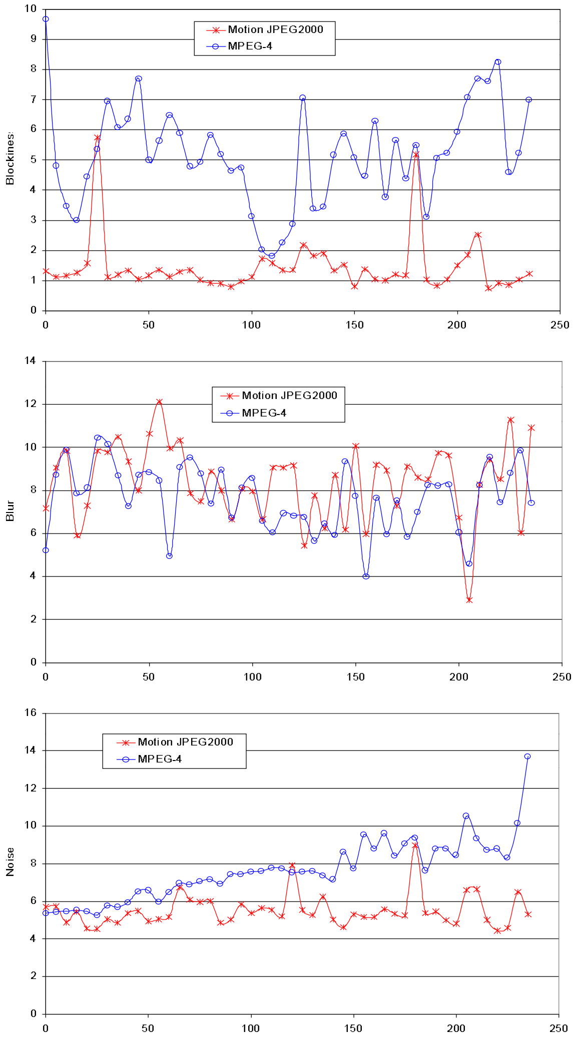

Figure 4: Perceptual results (top: blockiness, middle: blur, bottom: noise).

Figure 5: Sample frames (top: *Balloons*, bottom: *Suzie*, left: Motion JPEG2000, right: MPEG-4).

4.3. Complexity

Finally, Table 3 compares the computational complexity of Motion JPEG2000 and MPEG-4. The results have been obtained by profiling the software, and therefore depend on the optimization level of the respective codes. Note that the complexity results for MPEG-4 have been obtained with the Microsoft reference software, as it is much faster than the MoMuSys implementation.

	MJ2K	MP4
Encoder	88	390
Decoder	38	45

Table 3: CPU time ([ms] per frame).

We can see that MPEG-4 encoding is approximately 5 times more complex than MotionJPEG2000, while decoding is almost of equal complexity. MPEG-4 encoding is more complex because of motion estimation and compensation.

5. CONCLUSIONS

In this paper we have shown the strong performance of Motion JPEG2000. In particular we have shown its good error resilience for the transmission of video over error-prone wireless channels. More specifically, Motion JPEG2000 outperforms MPEG-4 by an average of 1.22 dB in the case of a bit error rate of 1e-4. In addition, Motion JPEG2000 offers other compelling features such as lower complexity, better scalability and low coding delay. We can therefore conclude that Motion JPEG2000 is very well suited for wireless applications.

ACKNOWLEDGEMENT

EPFL’s contribution to this work was partially supported by the European Project 2KAN. Please refer to http://www.2kan.org for more details about this project and its objectives.

REFERENCES

[1] ISO/IEC 15444-3:2002, “Information technology – JPEG2000 image coding system – Part 3: Motion JPEG2000”, 2002.

[2] D. Taubman and M. Marcellin, “JPEG2000: Image Compression Fundamentals, Standards and Practice”, Kluwer Academic Publishers, 2002.

[3] T. Ebrahimi and F. Pereira, “The MPEG-4 Book”, Prentice Hall, 2002.

[4] D. Santa-Cruz, R. Grosbois and T. Ebrahimi, “JPEG2000 performance evaluation and assessment”, Signal Processing: Image Communication, vol. 17, no. 1, pp. 113-130, January 2002.

[5] I. Moccagatta, S. Soudagar, J. Liang and H. Chen, “Error-Resilient Coding in JPEG-2000 and MPEG-4”, IEEE Journal on Selected Areas in Communications, vol. 18, no. 6, pp. 899-914 , June 2000.

[6] R. Talluri, “Error-resilient video coding in ISO MPEG-4 standard”, IEEE Commum. Mag., vol. 36, no. 6, pp. 112-119, July 1998.

[7] S. Winkler and F. Dufaux, “Video Quality Evaluation for Mobile Applications”, In SPIE Proc. Visual Communication and Image Processing, vol. 5150, Lugano, Switzerland, July 2003.

[8] ISO/IEC JTC1/SC29/WG1 WG1N2696, “JPWL Scope and Requirements”, March 2002.

[9] ITU-T Study Group 16, “WCDMA Error Patterns at 128 and 384 kbps”, Technical Report Q15-G28, 1999.

[10] http://www.kakadusoftware.com (v3.4).

[11] ISO/IEC JTC1/SC29/WG11 WG11N5550, “ISO/IEC 14496-7/DAM1 Optimized reference software for coding of audio-visual objects”, March 2003. http://mpeg.telecomitalialab.com/working_documents/mpeg-04/optimised_software/14496-7dam1.zip.

[12] http://www.genista.com.