MJPEG2000 TRANSMISSION OVER LOSSY CHANNEL

Ing. Milan BŘEZINA, Doctoral Degree Programme (2)
Dept. of Telecommunications, FEEC, BUT
E-mail: xbrezi06@stud.feec.vutbr.cz

Ing. Kamil BODEČEK, Doctoral Degree Programme (2)
Dept. of Telecommunications, FEEC, BUT
E-mail: xbodec00@stud.feec.vutbr.cz

Supervised by: Dr. Ivan Rampl

ABSTRACT

Obviously in wireless or mobile standards there is always a need to achieve a robust multimedia transmission. Applications require the best trade-off between Quality of services, bandwidth and delay. The core technology of Motion JPEG2000 targets an intra-frame-based coding system, which differs from the current MPEG (MPEG-1, MPEG-2, and MPEG-2) moving pictures standards, which provide both intra-frame and inter-frame coding. That provides better features such as easy editing and robustness with respect to an error prone applications as well as professional broadcast systems [1], [2]. This paper should be show that features in practise.

1 INTRODUCTION

The Motion JPEG2000 provides a set of features vital to many high-end and emerging video applications by taking an advantage of more modern technologies. In addition, the MJPEG2000 is a flexible format, permitting many uses, including editing, display, interchange, and streaming. The MJPEG2000's desirable capabilities support numerous markets and applications [3],[4]. In recent performance studies of the MJPEG2000, three aspects were examined: compression efficiency, error resilience, and image quality. These three aspects are chosen because they are crucial in visual communication. First of all, the compression ratio indicates the efficiency of a coding standard. Secondly, the robust transmission of compressed images and video sequences through error-prone media such as different coding schemes cause different kinds of image information losses. Therefore, it is worth studying how the image information loss incurred during transmission over a lossy channel by the various coding schemes affects the image quality [7].

2 SIMULATION PROGRAM

Kakadu software [6] was used to make compressions from image formats *bmp* and *ppm* to *jpc* or *jp2*. The Kakadu software will be shortly described in chapter 2.1. The whole program for simulations is written in Matlab and the most important part of the *M-files* is described in chapter 2.2. Our proposed block scheme is also depicted there. The goal of possible simulations is to analyse behavioural of transmitted video by lossy environment and to set encoder parameters and error resilient as well as possible. This program can simulate video transmission over lossy or lossless, one or two channels. The complete block diagram of this scheme is in Fig. 1:.

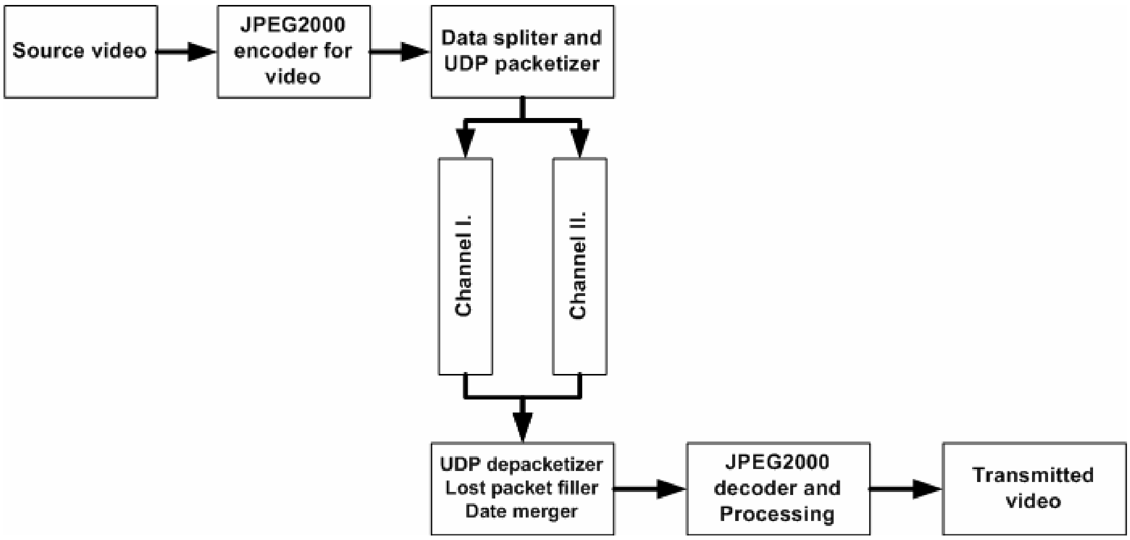

Fig. 1: *The full system structure*

The following part is focused on a short overview of the whole system work. First of all the video file has to be chosen. Then the error resilient and several properties of JPEG2000 encoder are being set. In the next step the data stream is separated into two parts and subsequently UDP packets are created from these parts. Parameters such as maximal bandwidth of one or both transmission channels are set as well. Furthermore, the behaviour of channel(s) is chosen. That means the parameters as a bit error rate, burst error rate, packet loss rate are set. After using the channel(s) it is important to depacketize the data stream and check if any packets are corrupted. Then the data steam should be merged into a single JPEG2000 image, and then decompressed it to *ppm* format and save this file to the hard drive. After video transmission, the processing is performed on the decompressed *ppm* file. This processing helps to make the final video as good as possible and it is the most important part of the proposed transmission scheme. Only the correct setting gives a good visual quality of transmitted video. When two transmitted videos are compared, one without processing and the other with using the processing, it can be easy seen, that if the processing is performed, the subjective visual video quality is better. It is because the human eyes are more sensitive to the damaged pictures within the video stream than to the same consecutive pictures. Our tests showed that this hypothesis is correct and the goal subjective visual video quality is better when the processing is employed.

2.1 KAKADU SOFTWARE

For our purposes the latest free Kakadu version 5.0 was chosen [5]. Kakadu is a complete implementation of the JPEG2000 standard, Part 2. Kakadu software has been written specifically with a variety of different types of applications in mind [6]. The core of Matlab program is based on these utilities. They can be used from command prompt only. It is not a disadvantage, because Matlab can use applications from command line as well.

Image compression: The compression of image files in a variety of formats is demonstrated by the "*kdu_compress*" utility. Image decompression: The streaming decompression of a JPEG2000 code-stream to an output image file is demonstrated by the "*kdu_expand*" utility. Transcoding between related representations: Many of the transcoding operations which are natural in the context of JPEG2000 are demonstrated by the "*kdu_transcode*" utility. Interactive rendering applications: Interactive (or non-linear) decompression and rendering to display are demonstrated by the "*kdu_show*" utility.

2.2 PROGRAM STRUCTURE

The Fig. 2: shows the last and the most important block. It includes several functions. The input parameters are already known values. The output is not only the resulting sequence in YUV format, but also the resulting text file, where information about several PSNR`s, the number of lost frames and the number of corrupted markers is saved.

The first section employs the *CheckError* function, which returns the number of corrupted SOP or/and EPH markers contained in a single frame. The markers found are counted and next compared with expected state. The distance value is returned via *ErSOP* or/and *ErEPH*.

In second section the *MakeYUV* function returns the YUV sequence. Among others it returns the number of lost frames and PSNR of each frame as well. A secondary product of that function is the YUV sequence without error and YUV sequence of first-degree processing. These sequences are useful for comparing the correct work of processing.

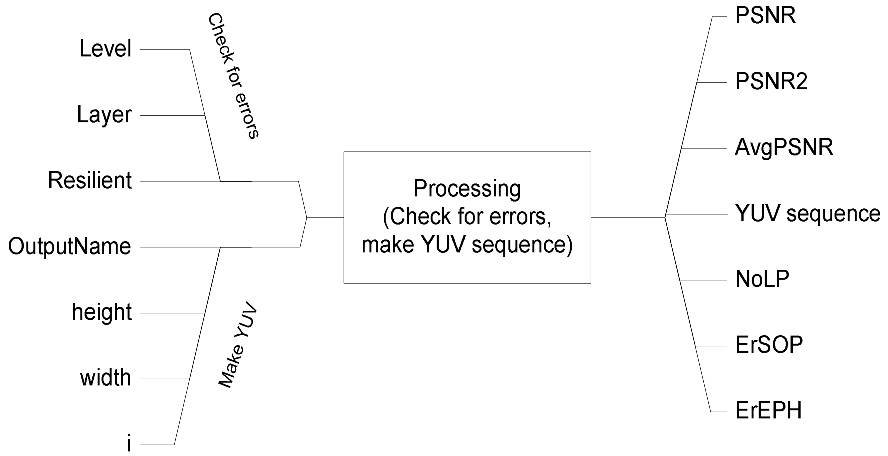

Fig. 2: *Processing and storage YUV sequence and Results*

First two buffers are created one to the PSNR calculation and one to save the previous frame. Subsequently it is checked if a new frame has been created (decoder created *ppm* file). If it is so, the identical size and length of the new picture and expected values are compared. In the case that all tests are positive, the PSNR of picture (is counted between two consecutive pictures) is counted and then compared with average values of five previous pictures. If the PSNR is in range, the new picture is saved into the video sequence. If at least one of the tests is negative or if any parameter is out of required range, the new picture is thrown away, and the preceding picture from the buffer is saved instead.

3 VIDEO SIMULATIONS

When a real-time video sequence is transmitting, there is no time for retransmission. We can say that two successive frames are very similar and if one is replaced by preceding frame, the final visual quality does not decrease so much as if the bad or corrupted frame was left in stream. Our simulations followed this idea. All simulations were done for three well-known test videos: Foreman, Mother and Daughter in the CIF format and Car phone in the QCIF format. The Car phone video has the length of 382 frames, the others only 300 frames. For one-channel scheme; the bandwidth is 432 Kb/s with 10^{-3} bit-error rate and for two-channel scheme; the bandwidths are 64 Kb/s with 10^{-6} bit-error rate and 332 Kb/s with 10^{-3} bit-error rate.

4 RESULTS OF REAL ENVIRONMENT

This simulation includes all possible types of errors and losses together. The following three tables (Tab. 1:, Tab. 2: and Tab. 3:) show results of a real time test. With certain inaccuracy it can be said all the results are similar. This type of simulation combines single, burst, packet and switch losses. EPH markers are very useful because the amount of non-created frames is smaller when resilient is used. The two-channel scheme has the smallest quantity of non-created frames of all sets. The remarkable paradox can be seen, while the number of non-created frames is smaller when two-channel scheme is used, the number of bad frames is vice versa higher. The total number of replaced frames is important as well. The number of non-created frames is higher for both channel schemes when no resilient is used. It is because without resilient non-corrupted frame can be repaired. The proposed algorithm has to recognize a bad frame (frame with low PSNR) and replace it. It is a protection against poor-quality frames. But any protection is total protection. There are several cases when the proposed algorithm cannot correctly recognize good or bad frames. In the first case program will a replace good frame by another preceding good frame, but then the video scene will lose time continuity. In second case the program leaves the bad frame in video sequence, while its PSNR is quite low. It will show as an unpleasant flash in the sequence.

Video QCIF – CarPhone	EPH markers		Without markers	
	1 ch	2 ch	1ch	2 ch
Number of replaced frames [%]	29,7	26,2	31,2	33,9
Number of non-created frames [%]	10,3	2,7	18,1	20,2
Number of bad frames (low PSNR) [%]	19,1	23,4	13,1	13,7

Tab. 1: *Result of CarPhone – 382 frames*

Video CIF – Forman	EPH markers		Without markers	
	1 ch	2 ch	1ch	2 ch
Number of replaced frames [%]	28,3	27,8	27,0	24,5
Number of non-created frames [%]	9,7	4,0	15,0	8,3
Number of bad frames (low PSNR) [%]	18,7	23,8	12,0	16,2

Tab. 2: *Result of Foreman – 300 frames*

Video CIF - MotherDaughter	EPH markers		Without markers	
	1 ch	2 ch	1ch	2 ch
Number of replaced frames [%]	26,7	25,8	27,3	27,0
Number of non-created frames [%]	9,5	1,8	13,3	7,7
Number of bad frames (low PSNR) [%]	17,2	24,0	14,0	19,3

Tab. 3: *Result of MotherDaughter- 300 frames*

In the scope of this work, the other simply types of simulations was performed. These are following: simple errors, burst errors, packet losses, switch losses and their combinations. The results of these simulation types cannot be included to this paper due their considerable size. It can be said, that most of the results are similar in respect of value ratio. Last type, real environment, is the worst case of all and its result is the worst as well.

5 CONCLUSION

This paper was focused on the JPEG2000 error-resilient tools, which can be used for transmission in mobile applications. Our purposed replacement by previous frame was used in simulations of real-time video transmission. The program can work also in image or video mode and its property and adjustment were shortly described in previous chapters. When using the proposed algorithm, the better subjective visual quality of the video can be seen. Although everything appears to be good, there is also one disadvantage. The proposed algorithm is a good idea how to remove a bad frame of sequence, but its outcomes depend on the video sequence used. Slow sequences need another adjustment as a fast sequence. The current PSNR of frame is compared with the fixed value and with average value of five previous frames. The number of bad markers is compared with fixed value as well. The probability, that a good frame is replaced or a bad frame is kept, is about 5 %. A better solution can be obtained with a more adaptive algorithm. If time allows, the new version of the program will be done in next work.

REFERENCES

[1] ITU-T Recommendation T.800 and ISO/IEC 15444-1, JPEG2000 Image Coding System

[2] ITU-T Recommendation T.800 and ISO/IEC 15444-3, Motion JPEG2000 Version 2

[3] Yu, W., Qiu, R., Fritts, J.: Advantages of Motion-JPEG2000 in Video Processing, Master Thesis, URL: http://www.enee.umd.edu/~weiyu/homepage/research/thesis.ppt

[4] Qiu, R., Yu, W., Fritts, J.: Motion JPEG2000 Video transmission over active network, 2002, URL: http://ccrc.wustl.edu/~jefritts/papers/ei03rq.pdf

[5] Taubman, D. S., Marcellin, M. W.: JPEG2000: Image compression fundamentals, standards, and practices, Kluwer, Norwell, MA, ISBN 0-7923-7519-x, USA, 2002

[6] Taubman, D.: Kakadu survey documentation, 2001,URL: http://www.kakadusoftware.com/Kakadu.pdf

[7] Pavlidis, G., Tsompanopoulos, A., Papamarkos, N., Chamzas, C.: JPEG2000 over noisycommunication channels thorough evaluation and cost analysis, Cultural & Educational Technology Institute, Greece , 2003