Journal of Theoretical and Applied Information Technology
10th November 2015. Vol.81. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

THE EXPERIMENTAL STUDIES OF THE JPWL TOOLS ABILITY TO CORRECT BURST ERRORS IN A NOISY CHANNEL WHEN TRANSMITTING VIDEO IN A JPEG 2000 FORMAT

SERGEY VASILIEVICH SKOROKHOD
Southern Federal University
Bolshaya Sadovaya Str., 105/42, Rostov-on-Don, 344006, Russia

ABSTRACT

The discussion is made on the issue of transmitting video in a JPEG 2000 format using the JPWL tools (Wireless JPEG 2000) for noise-immunity coding through a noisy channel under conditions of burst errors occurrence. The major problem is the restoration of the lost packet using the JPWL tools. It is set out to conduct an experimental study on the ability of JPWL tools to correct burst errors. The methodology of the study consists in modelling a transmission of the JPWL protected video composed of 1,000 frames of one and the same image with a size of 1,024 x 768 pixels. The variable parameters of the study are as follows: the Reed-Solomon codes applied for protection, the number of image tiles, and the RTP (Real-time Transport Protocol) packet loss ratio. A standard variant of protection and a combination of the standard variant with an interleaving algorithm are considered. The final results of the experiment are the average values of the percentage of fully and partially restored tiles of the number of tiles in a code stream. A software system developed for the conduction of study is described, which includes the JPWL encoder and decoder, the tools for partitioning code streams into RTP packets, the tools for packet loss modelling, and the tools for frame assembly from the RTP packets. A macro flowchart of the JPWL encoder and decoder functioning is described. A method for the code stream interleaving is suggested, which is intended for increasing its resistance to burst errors. During the study process, the two hypotheses were put forward, which were confirmed by the experimental results obtained. The first hypothesis is that the standard JPWL tools are not able to correct burst errors. The second hypothesis is that the standard JPWL tools used in conjunction with the interleaving algorithm can recover burst errors. A side effect of the study is the conclusion on a slight improvement of the recoverability of tiles with the increasing number of tiles in a code stream.

Keywords: *JPEG 2000, JPWL, Burst Errors, Noise-Immunity Coding*

1. INTRODUCTION

This paper deals with the general problem of video transmission in the JPEG 2000 format [1] through a noisy channel, in particular, over wireless networks. The transmission is related to forwarding network packets that carry fragments of an encoded video stream. Due to the noise contamination of a channel, some packets can be corrupted and discarded. The consequence of such burst errors is a lack of adjacent fragments of video data in a receiver, which either complicates their decoding or renders the decoding procedure impossible. In order to protect a JPEG 2000 code stream during the transmission over wireless networks, a specification of ITU T.810 [2] was developed. It describes a set of tools applied for a single image, which are called JPWL. Since the JPEG 2000 video is a sequence of mutually independent images [3], JPWL is also used to protect video data.

The purpose of this work is to conduct experimental studies on the ability of JPWL tools to correct burst errors. By a burst error is meant the loss of at least one network packet with data when transmitting it from a source to a receiver [4]. In order to carry out this study, a specialized software system is developed that is capable of performing JPWL encoding and decoding, partitioning a JPWL code stream into a sequence of network packets, and reassembling a JPWL code stream from the network packets with the modeling of their losses. This paper also considers the method of interleaving that can significantly improve the ability of standard tools to correct the burst errors, which is also a subject of the experimental study.

Journal of Theoretical and Applied Information Technology
10th November 2015. Vol.81. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2. STUDY METHODOLOGY

1.1. The studied structure of a code stream

The study methodology is based on the set of JPWL protection tools designed to protect the code stream of a video frame [5]. When encoding in the JPEG 2000 format, a frame can be divided into several rectangular fragments (tiles), each of which is coded separately. The structure of a code stream is presented in Figure 1 [6].

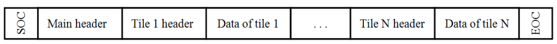

Fig. 1. The Structure Of A JPEG 2000 Code Stream

The SOC and EOC markers indicate the beginning and the end of a particular frame. The main header contains the image data as a whole: size, color scheme, number of tiles, etc. The tile header contains information on the rectangular image fragment including its grid lock. The tile data is actually an encoded image fragment.

The JPWL tools can use two methods of protection [2]:

- The *CRC-16* or *CRC-32* checksums that can only allow detecting the availability of errors in the data when using a minimum amount of parity codes.
- The Reed-Solomon codes [7] of *RS(n,k)* (*RS*-codes) – intended not only for detecting but also for correcting the *(n-k)/2* corrupted data. Here, *n* is the length in bytes of a code word consisting of the protected data fragment and parity codes, and *k* is the number of bytes of the protected data. The *RS(n,k)* code allows correcting up to *(n-k)/2* of corrupted bytes in a code word. The ITU T.810 specification provides for the usage of an entire family of *RS*-codes from the "weakest" ones of *RS(37,32)* up to the "strongest" ones of *RS(128,32)* and *RS(160,64)*.

Since the checksums cannot be used to protect against burst errors, the present paper discusses the use of RS-codes only.

A code stream is protected by adding thereto special segments. There are four types of segments provided.

- EPC — the error protection capability. It is inserted only into the main header. Its mission is to inform a decoder that a code stream includes JPWL tools. It contains the description of the applied protection methods and, if necessary, their parameters.
- EPB — the error protection block. It actually implements the data protection using the *RS*-codes. It is inserted into the main header and the tile headers. It contains the applied protection parameters and the redundant parity codes.
- ESD — the error sensitivity descriptor. It determines the degree of sensitivity to errors in the code stream fragments. The more a code stream fragment contributes to the image being formed, the greater the sensitivity is.
- RED — the residual error descriptor. It is used only in the headings of a decoded code stream. It describes the areas, which are either fully recovered or have not been recovered by the JPWL descriptor and contain residual errors.

The ESD segment is used only by the JPWL encoder in order to implement the better protection of the most valuable areas of a code stream. It does not carry any useful information for the JPWL decoder, therefore significantly increasing the sizes of headings and of the entire code stream in general. In this regard, during the video transmission, it is advisable to abandon the use of the ESD segment [8, 9], and in the present study, this segment is not used.

1.2. The method and program of the study

The methodology of the study consists in modelling a transmission of the JPWL protected video composed of 1,000 frames of one and the same image with a size of 1,024 x 768 pixels, which was previously encoded by the JPEG 2000 encoder of the OpenJpeg 2.1.0 system [10]. The resulting code stream is loaded into the research software system developed by the author, in which he consistently passes the following stages of processing:

- The JPWL encoding.
- The partitioning into RTP packets, which correspond to the RFC 5371 specification [11, 12].
- The modelling of packet losses according to the specified percentage of losses.
- The assembly of a code stream from the remained packets.
- The JPWL decoding and the calculation of the decoding statistics.

The constant parameters in the study are the method of headings protection and the size of an RTP packet. In order to protect the main header and the tile headers, the RS-codes are applied, which

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

are predefined in the T.810 specification. The RTP packet size equals to 1,024 bytes.

The variable parameters of the study are the types of RS-codes being applied for the data protection, the packet loss ratio, and the number of tiles, into which an image is partitioned (it is determined by the selected size of a tile). The study program is presented in Table 1.

Table 1. Study Program

The number of tiles (tile size)	The *RS*-codes for data protection	The RTP packets loss ratio
12 (256 x 256) 48 (128 x 128) 192 (64 x 64)	*RS(37,32)*, *RS(64,32)*, *RS(96,32)*, *RS(128,32)*	1, 3, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50

The statistical results of the decoding are the number of fully and partially recovered tiles of a code stream.

It should be noted that the most important part of a code stream is its main header. Therefore, for its added protection, the excessive duplication is usually applied to the packets containing its fragments [13], due to which, in this study, the burst errors will not be applied to the packets of the main header.

2. MATERIALS AND METHODS

2.1. The structure and functioning of the software system

The dedicated software system was specially developed for this study, which consists of two modules: the preparatory one and the modelling one. The scheme of the system functioning is shown in Figure 2.

The input data for the preparatory module is the JPEG 2000 code stream preliminary generated using the OpenJpeg 2.1.0, which comprises 12, 48 or 192 tiles in compliance with the study program as well as the set of study parameters, in which the coding mode (with or without the use of interleaving) and the percentage of discarded packets are specified. The preparatory module performs the JPWL coding and the interleaving, if required, partitions the obtained code stream into the RTP packets in compliance with the RFC 5371 specification, and downloads into intermediate files the obtained sequence of packets, their parameters (quantity, length, etc.), and the study parameters.

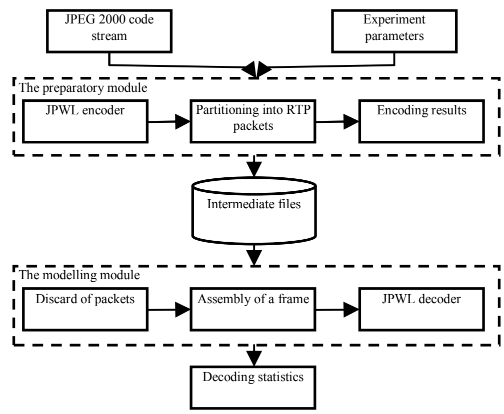

Fig. 2. The Research Software System

From the intermediate files, the modelling module uploads the prepared sequence of packets and the study parameters. Further, the cyclical iterations of the packet discard, the assembly of a frame and its processing by the JPWL decoder are carried out. The decoder counts the number of fully and partially recovered tiles. After executing 1,000 of similar iterations, the arithmetic mean of these values is calculated, which characterizes the ability of the decoder to recover the code stream at the given parameters of encoding and at the specified level of burst errors.

The JPWL encoder and decoder are the main units of this system. The present study uses the encoder and decoder libraries developed by the author for the system of video streaming [14, 15, 16, 17].

2.2. The JPWL encoder

A generalized flowchart of the JPWL encoder operation is shown in Figure 3. The input data are in fact presented by the JPEG 2000 code stream and the parameters of the applied JPWL protection tools. The encoder performs the markup of the main header and of each subsequent tile in the output buffer. At the same time, the data on the parameters of JPWL markers and their locations are recorded in the markup table of the output buffer. After finishing the passage through the output buffer, the length of the output stream is calculated as the sum of the input stream length and the lengths of all markers added thereto. The obtained length value is put into the markup table into the parameters of the EPC marker. Further, the input data and the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

parameters of the JPWL markers are copied to their places in the output stream. The last stage of encoding is the calculation and input of the EPB markers' parity codes into the output stream. After this stage, the JPWl code stream is generated that meets the requirement of the backward compatibility, i.e. the JPEG 2000 decoder is capable of decoding this stream if it is not corrupted. When applying the algorithm of interleaving, an additional interleaving is performed, which results in the JPWL code stream that meets the requirement of the backward compatibility with the extension: the JPEG 2000 decoder is capable of decoding such stream only at the availability of the JPWL decoder.

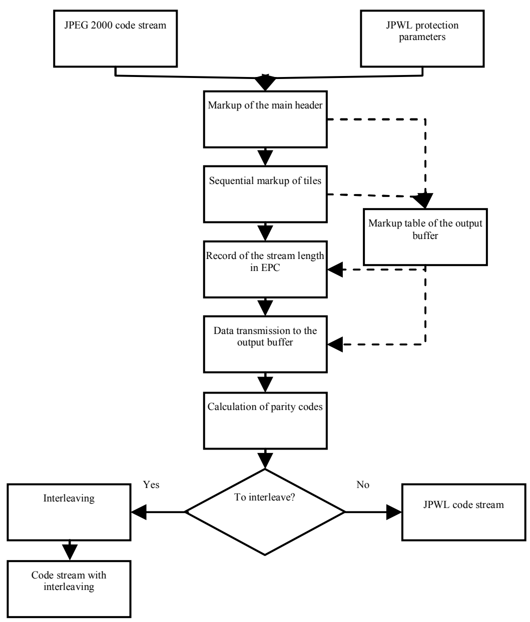

Fig. 3. The JPWL Encoder Operation Flowchart

2.3. The JPWL decoder

A generalized flowchart of the JPWL decoder operation is shown in Figure 4. The JPWL decoder decodes the preassembled frame consistently analyzing the main header and the tile headers. If it is impossible to recover the main header, the whole frame is discarded. If the main header is successfully recovered, then the EPC segment analysis makes it possible to determine whether the interleaving of a code stream is applied or not. If the interleaving is applied, the decoder at first performs the inversion and only afterwards proceeds to analyze the tiles. If it is impossible to recover the tile heading, the whole tile is omitted, and the heading of the next tile is searched for. However, the omitted fragment is marked in the markup table of the input buffer as a corrupted one. In the event of a successful recovery of the main header or the tile header, the markup table is completed with the parameters and locations of all found JPWL markers. The recovery of the tile data is carried out only in the event of a successful recovery of its heading. At the same time, a part of the tile data can prove to be corrected and the other part — non-corrected. In this case, the entries are made into the table of RED intervals, which describe the beginning and the end of both the corrected and the non-corrected data fragments.

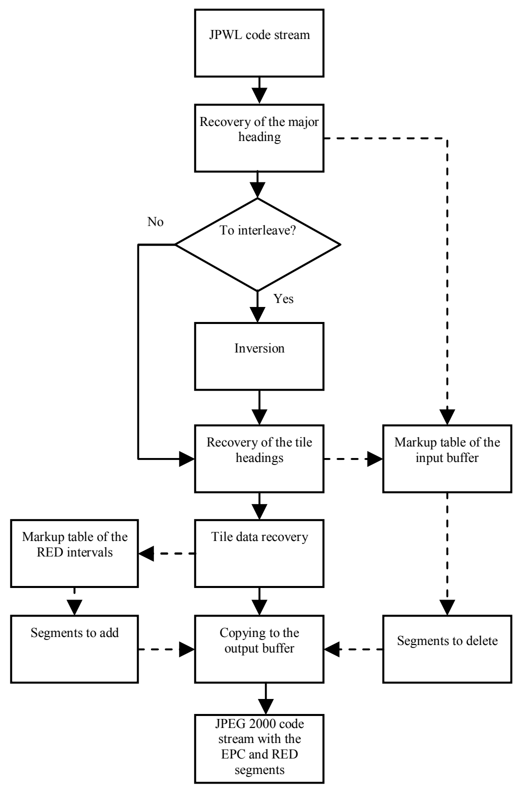

Fig. 4. The JPWL Decoder Operation Flowchart

Upon completion of viewing the input buffer, the data are copied into the output buffer, wherein all the JPWL input markers are omitted. If the table of RED intervals contains entries, the EPC segment is inserted into the main header indicating the presence of the RED segments in the code stream, and the headings of the partially recovered tiles are supplemented with the RED segments describing the addresses of the recovered or non-recovered fragments.

The statistics of the JPWL decoder is calculated based on the results of the recovery of tile headings and data. If, after the process of decoding, the tile

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

heading does not contain a RED segment, such tile is considered to be totally recovered. If a RED segment is present, the tile is considered to be recovered partially. The availability of the RED segment says nothing about the data quality in the partially recovered tile. Therefore, this study does not address this issue.

2.4. The method of interleaving

The JPWL encoder and decoder support both the standard method of protection corresponding to the T.810 specification and the algorithm of interleaving not relating to this specification [18]. The essence of the algorithm is that the main header remains unchanged, and the rest part of the code stream is subjected to the inversion. The main header contains an EPC segment, in which the total length of the code stream is recorded. The difference of the total length and the length of the main header determines the length of data that are to be inversed. These data are considered as a rectangular matrix filled in "by rows". The data in this matrix are inversed so that it could be filled in "by columns".

Let us assume that L is the total length of a code stream, M is the length of the main header. Then, according to [19], the number of columns and rows of the matrix is calculated by the following formula:

$$N_C = \left\lceil \sqrt{L-M} \right\rceil,\ N_R = \left\lceil \frac{L-M}{N_C} \right\rceil.$$

It should be noted that as a result of such inversion, the actual length of the code stream is somewhat increased on the side of the encoder, since the matrix can include up to NR-1 of "extra" elements. The JPWL encoder and decoder should consider this fact.

The difference of the proposed interleaving method from the method described in [19] is that in this case, the interleaving is applied to the entire code stream except for the main header. The specification [19] suggests the interleaving of the parity codes only that are recorded in the EPB segments.

3. THE EXPERIMENTAL STUDY

3.1. Hypotheses and criteria

The study verifies two main hypotheses.

Hypothesis 1. The standard JPWL tools corresponding to the T.810 specification are not able to correct the burst errors. The criterion for confirming this hypothesis is the inverse linear dependence of the packet loss ratio and the total number of fully and partially restored tiles.

Let us analyze the standard method of protection using the RS-codes. The RS(n,k) code word comprises two parts located in different portions of the code stream. The first part consists of the redundant parity codes with the length of n-k bytes, which are located in the main header or in the tile headers. The second part consists of the k information bytes protected from corruption. If a packet is lost, three variants are possible [20].

1. Both parts of the code word are lost — the recovery is impossible.
2. The redundant parity codes are lost — the recovery is impossible.
3. Only the information bytes are lost — the recovery is possible only if $k \le (n-k)/2$. This condition is met only by the *RS(160,64)*, *RS(80,25)*, *RS(40,13)* codes applied to protect the tile headings, and by the *RS(96,32)*, *RS(112,32)*, *RS(128,32)* codes applied to protect the tile data. The remaining 14 codes predefined by the standard are not able to perform a recovery.

The analyzed in the study methods of protection by the RS(37,32) and RS(64,32) codes are not able to correct data in neither of the above mentioned variants. When applying the RS(96,32) and RS(128,32) codes, the data correction is possible in the variant 3, though this will cause a significant increase in the tile headings, which makes them much more vulnerable and can neutralize a possible positive effect.

Hypothesis 2. The standard JPWL tools applied in conjunction with the algorithm of interleaving can recover the burst errors. The recovery ability depends on the used RS-codes and on the packet loss ratio. The criterion for confirming this hypothesis is the nonlinear dependence of the packet loss ratio and the total number of fully and partially restored tiles.

3.2. The test of hypothesis 1

In order to verify the hypothesis 1, let us conduct an experiment 1. The experiment is conducted on a JPEG 2000 code stream, which comprises 48 tiles, and to which the standard protection is applied. The applied RS-codes and the modelled percentage of RTP packet losses are specified in the study program. The results of the experiment are given in Table 2 with rounding to integer values. The size of the JPWL code stream, the mean percentage of the fully and partially recovered tiles, and the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

correlation coefficient of the sum of the recovered tiles percentage and the packet loss ratio are specified for each type of RS-codes.

Table 2. Results Of The Experiment 1

Packet losses, %	RS(37,32) 1,025,691 bytes		RS(64,32) 1,767,653 bytes		RS(96,32) 2,647,205 bytes		RS(128,32) 3,526,937 bytes	
	Fully	Partially	Fully	Partially	Fully	Partially	Fully	Partially
1	81	18	71	28	69	30	58	40
3	53	44	34	63	35	61	21	76
5	35	60	18	77	16	79	8	85
10	12	78	4	86	4	86	1	87
15	5	81	1	84	1	84	0	81
20	2	78	0	81	0	80	0	76
25	1	75	0	76	0	75	0	71
30	0	69	0	70	0	70	0	65
35	0	64	0	65	0	64	0	60
40	0	59	0	59	0	61	0	54
45	0	55	0	55	0	56	0	50
50	0	50	0	49	0	51	0	43
		-0.999		-0.999		-0.999		-0.999

The correlation coefficients given in the bottom line demonstrate an absolute inverse dependence of the sum of the recovered tiles percentage and the packet loss ratio. This result completely confirms the hypothesis 1. Moreover, the expectations for that the RS(96,32) and RS(128,32) RS-codes will give a better result as compared to the RS(37,32) and RS(64,32) codes were not justified due to the excessive increase in the tile headings when applying these codes.

Within the framework of verification of the hypothesis 1, the experiments 2 and 3 were also conducted, in which the JPEG 2000 code stream consisting of 12 and 192 tiles, respectively, was used. The rest conditions corresponded to the experiment 1. The results of the experiments 2 and 3 are shown in Tables 3 and 4.

Figures 5–7 demonstrate dependency graphs of the cumulative percentage of fully and partially restored tiles (ordinate) on the percentage of burst errors (abscissa) for the experiments 1–3. The obtained dependences completely correspond to the criterion that confirms the hypothesis 1 in all experiments. Furthermore, it can be concluded that alongside with the increase in the number of tiles, the percentage of their recovery also increases. This comes from the fact that the higher the quantity of tiles is, the smaller their size is, which reduces the probability of damage of a certain tile, especially of its heading, at a burst error occurrence. On the contrary, the lower the quantity of tiles is, the bigger their size is and the more vulnerable for a burst error they become.

Table 3. Results of the experiment 2

Packet losses, %	RS(37,32) 1,003,010 bytes		RS(64,32) 1,733,367 bytes		RS(96,32) 2,599,119 bytes		RS(128,32) 3,464,911 bytes	
	Fully	Partially	Fully	Partially	Fully	Partially	Fully	Partially
1	45	54	23	76	25	72	13	83
3	10	87	3	92	2	91	1	87
5	2	91	1	92	0	88	0	82
10	0	90	0	83	0	75	0	68
15	0	87	0	77	0	63	0	54
20	0	79	0	71	0	55	0	45
25	0	76	0	62	0	49	0	36
30	0	70	0	55	0	37	0	27
35	0	63	0	49	0	34	0	21
40	0	60	0	42	0	25	0	16
45	0	55	0	39	0	20	0	11
50	0	51	0	32	0	17	0	7
		-0.998		-0.997		-0.988		-0.979

Table 4. Results of the experiment 3

Packet losses, %	RS(37,32) 1,003,010 bytes		RS(64,32) 1,733,367 bytes		RS(96,32) 2,647,205 bytes		RS(128,32) 3,526,937 bytes	
	Fully	Partially	Fully	Partially	Fully	Partially	Fully	Partially
1	94	5	90	9	90	9	86	13
3	83	14	73	24	74	23	63	34
5	73	22	60	35	60	35	47	48
10	52	38	36	54	36	54	23	67
15	38	47	21	64	22	64	11	74
20	27	53	13	68	13	68	5	75
25	19	56	7	68	7	68	3	72
30	13	57	4	66	5	66	1	68
35	9	56	3	63	2	63	1	64
40	6	53	2	58	1	58	0	60
45	4	51	1	54	1	54	0	55
50	3	47	1	49	0	50	0	50
		-0.999		-0.999		-0.999		-0.999

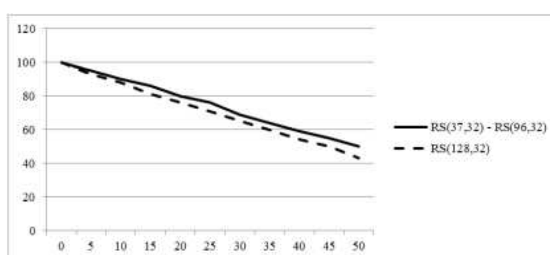

Figure 5. The Graph Of The Experiment 1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

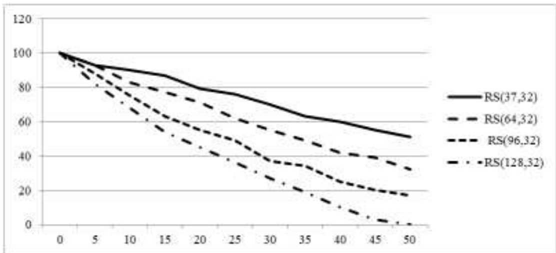

Figure 6. The Graph Of The Experiment 2

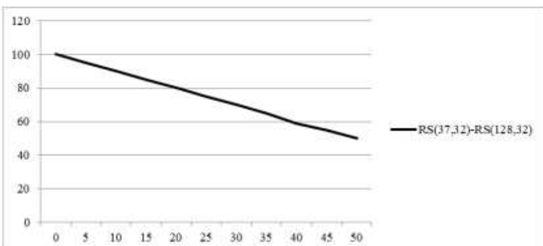

Figure 7. The Graph Of The Experiment 3

3.3. The test of hypothesis 2

In order to verify the hypothesis 2, let us conduct an experiment 4. Its input data correspond to the experiment 1, but the standard method of protection is combined with the interleaving algorithm. The results of the experiment are given in Table 5.

An analysis of Table 5 confirms the hypothesis 2. Indeed, at the level of packet losses being up to 25%, any RS-codes can recover almost all tiles, only the distribution of the fully and partially recovered tiles is changed. Alongside with the further growth of the packet losses, the percentage of the recovered tiles starts to decrease. Experiment 4 demonstrated another explicit dependence — when applying "longer" RS-codes, the quality of the recovery improves, primarily due to a greater percentage of the fully recovered tiles.

Table 5. Results Of The Experiment 4

Packet losses, %	RS(37,32) 1,027,949 bytes		RS(64,32) 1,770,615 bytes		RS(96,32) 2,649,924 bytes		RS(128,32) 3,529,845 bytes	
	Fully	Partially	Fully	Partially	Fully	Partially	Fully	Partially
1	39	61	100	0	100	0	100	0
3	0	100	100	0	100	0	100	0
5	0	100	100	0	100	0	100	0
10	0	100	98	2	100	0	100	0
15	0	100	28	72	100	0	100	0
20	0	100	0	100	85	15	100	0
25	0	98	0	99	7	91	83	15
30	0	92	0	92	0	92	7	84
35	0	72	0	73	0	72	0	70
40	0	37	0	39	0	39	0	35
45	0	11	0	11	0	11	0	9
50	0	1	0	1	0	1	0	1
	-0.86		-0.86		-0.86		-0.86	

Figure 8 demonstrates a dependency graph of the cumulative percentage of fully and partially restored tiles (ordinate) on the percentage of burst errors (abscissa) for the experiment 4. The obtained dependences completely correspond to the criterion that confirms the hypothesis 2.

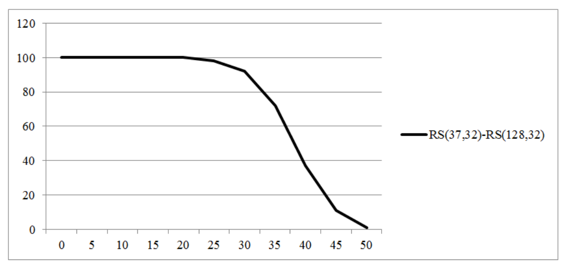

Figure 8. The Graph Of The Experiment 4

Within the framework of this study, the experiments 5 and 6 were conducted, the conditions of which coincided with the conditions of the experiment 4, but the JPEG 2000 code stream consisting of 12 and 192 tiles, respectively, was used. The results of these experiments largely coincide with Table 5 and Graph 8 with some deterioration at 12 tiles and improvement at 192 tiles.

4. SUMMARY

This study was conducted on a basis of the software system developed in C language within the Microsoft Visual Studio 2008 programming software. The obtained results can be used to choose the optimal JPWL parameters depending on the characteristics of a data transmission channel.

The focus of the further research is an experimental study of the impact caused by the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

burst errors on the quality of a decoded image in the JPEG 2000 format. In order to conduct this study, it is required to develop a JPEG 2000 decoder capable of decoding a partially recovered code stream, in which some tiles may be missing, and in the headers of the available tiles, the segments of the residual RED error can be present. Unfortunately, the known available decoders, such as JasPer, OpenJpeg [21], cannot decode such code stream.

5. CONCLUSIONS

According to the results of the conducted experiments, the following conclusions can be made.

1. A set of standard JPWL tools corresponding to the T.810 specification is insufficient for the effective correction of burst errors.
2. A combination of the standard JPWL tools with the algorithm of interleaving suggested in this work allows to successfully correct the burst errors.
3. An increase in the number of tiles slightly enhances the immunity of a code stream to the burst errors.

7. ACKNOWLEDGEMENT

The work is performed in the framework of execution of the basic part of the government assignment in the field of scientific activity (Project No. 3442 "The information and algorithmic support of digital control systems, autonomous high-precision navigation, and technical vision for the promising air vehicles: the development of the theoretical foundations of designing, algorithms, and methods of efficient and reliable software implementation, and the use of the high-performance computing infrastructure for experimental modeling").

The author expresses special gratitude to his colleagues who provided invaluable assistance in conducting this study: Khusainov, N.Sh., Kravchenk, P.P., Drozdov, S.N., Zhiglatyi ,A.N.

REFERENCES:

[1] Taubman, D.S. and M.W. Marcellin, 2002. JPEG2000: Image Compression Fundamentals, Standards and Practice. Boston: Kluwer Academic Publishers, pp: 773.

[2] ITU-T Recommendation T.810, 2007. Information Technology JPEG2000 Image Coding System: Wireless. Geneva: ITU, pp: 60.

[3] Acharya, T. and P. Tsai, 2005. JPEG2000 standard for image compression: concepts, algorithms and VLSL architectures. Hoboken: John Wiley & Sons, Inc., pp: 274.

[4] Skorokhod, S.V. and A.P. Stepanova, 2014. The Issues on the Synchronization of a Transmitter and Receiver during the Process of Video Streaming. In Collected Materials of the All-Russian Scientific-Practical Conference "Management of Intellectual Property and Innovation Activity of Economic Entities, Commercialization of Technologies – 2014", Taganrog, pp: 122-125.

[5] Skorokhod, S.V. and A.S. Artiukhova, 2014. The Specific Features of the JPWL System Implementation for the Broadcast Video Streaming. In Collected Materials of the All-Russian Scientific-Practical Conference "Management of Intellectual Property and Innovation Activity of Economic Entities, Commercialization of Technologies – 2014", Taganrog, pp. 131-134.

[6] ITU-T Recommendation T.800, 2003. Information Technology – JPEG2000 Image Coding System: Core Coding System. Geneva: ITU, 212 p.

[7] Morelos-Saragosa, R., 2005. The Art of Noise-Immunity Coding. Methods, Algorithms, Application. Moscow: Tekhnosfera, pp: 320.

[8] Skorokhod, S.V., P.P. Kravchenko and N.S. Khusainov, 2004. Some aspects of JPWL implementation for streaming video. In International Conference "Engineering & Telecommunication En&T 2014", November 26-28, ,Book of Abstracts, Moscow-Dolgoprudny, MIPT, pp: 135-136.

[9] Skorokhod, S.V. and A.P. Stepanova, 2015. Some Technological Aspects of the Network Interaction between a Transmitter and a Receiver during the Process of Video Streaming. In Proceedings of SFedU. Technical Sciences, 4(165): 171-181.

[10] An open-source JPEG 2000 codec written in C, 2015. In Network Working Group RFC 5371, RTP Payload Format for JPEG 2000 Video Streams, The Internet Security (IETF), pp: 31. Date views 01.09.2015 http://www.openjpeg.org.

[11] Network Working Group RFC 5371, 2008. RTP Payload Format for JPEG 2000 Video Streams. Sony. 31 p.

[12] Network Working Group RFC 3550, 2003. RTP: A Transport Protocol for Real-Time Applications. Schulzrinne, Blue Coat Systems Inc., pp: 89.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[13] Drozdov, S.N., 2014. Implementation of the JPEG 2000 Encoding Algorithms for the System of Video Data Transmission in Real Time. In International Conference "Engineering & Telecommunication En&T 2014", November 26-28, Book of Abstracts, Moscow-Dolgoprudny: MIPT, pp. 67-69.

[14] Khusainov, N.Sh., S.N. Drozdov, S.V. Skorokhod, A.A. Zhiglatyi and P.P. Kravchenko, 2014. The specific Features of Designing and Software Implementation of the System of Video Streaming on the Basis of the JPEG 2000 Family Standards. In International Conference "Engineering & Telecommunication En&T 2014", Book of Abstracts, Moscow, pp: 137-139.

[15] Drozdov, S.N., A.A. Zhiglatyi, P.P. Kravchenko, S.V. Skorokhod and N.Sh. Khusainov, 2014. On the Experience in Implementation of the System of Video Streaming in the JPEG 2000 Format and on the Prospects of the JPEG 2000 Standard Application for the UAV-Borne Transmission of Video and Multispectral Data. In Proceedings of SFedU, Technical Sciences, 7(156): 161-170.

[16] Drozdov, S.N., A.A. Zhiglatyi, P.P. Kravchenko, S.V. Skorokhod and N.Sh. Khusainov, 2014. On the Experience of Practical Implementation of the JPEG 2000 Family Standards at the Development of a Software System for Video Streaming in Real Time under Conditions of a Limited Network resource. In Proceedings of the Ninth All-Russian Scientific-Practical Conference "Future Systems and Control Problems" and the Fourth Youth School-Workshop "Management and Processing of Information in Technical Systems", Taganrog, pp: 459-472.

[17] Zhiglatyi, A.A. and P.P. Kravchenko, 2014. Managing the Coefficient of Compression of the Sequence of Images during the JPEG 2000 Encoding for Transmission via a Communication Channel with a Limited Bandwidth). In International Conference "Engineering & Telecommunication En&T 2014", November 26-28, Book of Abstracts, Moscow-Dolgoprudny: MIPT, pp: 72-75.

[18] Skorokhod, S.V., P.P. Kravchenko and N.Sh. Khusainov, 2014. An Analysis of the Implementation of Tools for the Stream Video Protection in the JPEG 2000 Format against the Errors in a Data Transmitting Channel for the On-Board Systems of Video Streaming. In Proceedings of SFedU, Technical Sciences, 8(157): 219-226.

[19] ITU-T Recommendation T.810. 2012. Information Technology JPEG 2000 image coding system: Wireless. Amendment 1: IP based wireless networks. Geneva: ITU-T Series T: Terminal for Telematic Services, pp: 6.

[20] Drozdov, S.N., A.N. Zhiglatyi, P.P. Kravchenko, V.N. Lutai, S.V. Skorokhod and N.Sh. Khusainov, 2014. The JPEG 2000 Standard: the Basic Algorithms, Examples of Implementation, and Application Prospects. Rostov-on-Don: SFedU Publishing House, pp: 255.

[21] Adams, M. and F. Kossentini, 2000. JasPer: A Software-Based JPEJ-2000 Codec Implementation. In Proc. of IEEE International Conference on Image Processing, Vancouver, BC, Canada, Oct. 2000, vol. 2, pp. 53-56. DOI: 10.1109/ICIP.2000.899223.