JPEG2000 OVER NOISY COMMUNICATION CHANNELS THE COST ANALYSIS ASPECT

G. Pavlidis, A. Tsompanopoulos, N. Papamarkos, C. Chamzas

Cultural & Educational Technology Institute, 58 Tsimiski Street, Xanthi 67100, Greece

ABSTRACT

In this paper we present the behavior of JPEG2000 coding scheme over noisy or congested communication channels and highlight the cost analysis aspect. Two error schemes are considered including bit errors and packet dropping effects. Two bit error methods are used, consisting of flipping or dropping the bits, and various packet sizes are put to the test of packet dropping. Extensive performance results are presented and the overall cost analysis is emphasized.

1. INTRODUCTION

One of the problems in engineering a packet switched network carrying both non-bursty delay-sensitive traffic (voice, video) and highly bursty delay-tolerant traffic (computer data, image data) is the congestion problem [1].

Because digital bitmap representations of images require large numbers of bits, data compression techniques are important for efficient transmission. Standard lossless compression methods, such as the lossless DCT-based JPEG or the JPEG-LS and JBIG coders, provide with compression ratios of about 2:1 on the average. Unfortunately, such algorithms do not have the ability to allow packet dropping by the network. Hence, when a congested facility drops a packet containing compressed image data, the rest of the image is destroyed, unless the end-user is employing an end-to-end receive - acknowledge - transmission - repeat mechanism. Such a protocol saves the transmitted information, but ultimately only makes matters worse for the already congested network as it further increases traffic, not to mention the additional disadvantage of increasing transmission delay. Thus, to be effective as a congestion relieving mechanism, packet dropping must be allowed to be with the knowledge and blessing of the end-user. Presumably such a user would be given pricing advantages for the droppable information, since this information is delivered only when the network is idle. Similar problems appear when noisy communication channels carrying delay-sensitive image data change bits of information.

JPEG2000, the new coding standard, comes to fulfill such requirements of progressive coding while providing with error resilience. Several papers and publications consider the performance of this coder in noisy environments in order to compare the scheme with the existing ones [2-5]. In this work, we present the results of using JPEG2000 coder in error resilient mode of operation with Layer-Resolution-Component-Position (LRCP) priority, considering the overall effect of different error models. The outcome of these results is an overall communication channel cost analysis that can be used by providers to impose fee policies and users to evaluate their provided services.

2. JPEG2000 AND ERROR RESILIENCE

JPEG2000 uses a variable length coder (the MQ arithmetic coder) to compress the quantized wavelet coefficients, and thus, is prone to channel or transmission errors. A bit error results in loss of synchronization at the entropy decoder and the reconstructed image can be severely damaged. To improve the performance of transmitting compressed images over error prone channels, error resilient bit stream syntax and tools are included in the standard. These error resilience tools deal with channel errors using the following approaches [5,7,8]:

- data partitioning and resynchronization,
- error detection and concealment, and
- Quality of Service (QoS) transmission based on priority

A summary and description of the error resilience and how it can be achieved at the entropy coding level and at the packet level can be found in [5]. Typical decoded image quality values describing the performance of JPEG2000 coder in noisy communication networks can be found in [2].

For the purposes of our work we have encoded the three standard test images ‘woman’, ‘café’ and ‘bike’, using the public available kakadu system version 3.0 [9].

For resilient encoding we use the kakadu command:

```
kdu_compress -i $1.pgm -o $1.jpc
-rate -,.1,.25,.5,.75,1,1.5,2 -full Cuse_sop=yes Cuse_eph=yes
Creversible=yes
Cmodes="RESET|RESTART|ERTERM|SEGMARK"
```

where $1 represents the image filename, while resilient decoding is guaranteed by using the command:

```
kdu_expand -i $1.jpc -o $1.pgm
-resilient_sop
```

The produced encoded images are stored in codestreams with progressive-by-layer reconstruction up to lossless (using intermediate rates of 0.1, 0.25, 0.5, 0.75, 1, 1.5, 2 bpp), including error resilience markers. The extra markers for the error resilience impose an overhead of about 1%, as expected and referenced by many other works on this subject [2-4,6].

The encoded images have to pass through a virtual noisy communication channel and to be decoded in order to evaluate the robustness of the encoder and to produce cost analysis data. The channel is simulated by a computer program, which hits the encoded codestream using two basic error schemes:

1. the codestream is hit in bit level (noisy channel), using two probability parameters:
    i. the error probability
    ii. the burst probability
2. the codestream is hit in packet level (congested network)

In scheme 1, the error probability p_e is the well known error rate with typical values of 10^{-6}, 10^{-5}, 10^{-4}, while the burst probability imposes an additional error probability simulating bursty noise or outage periods, common to real-life telecommunication systems. Burst errors applied are of rates $p_b=1-q$, where q is the burst factor from 10^{-7} to 10^{-1} in 10^{1} increments, with the minimum factor corresponding to higher burst probability. Thus, $p_e=10^{-6}$ and $q=10^{-2}$ would mean that with probability 10^{-6}, we flip or drop a continuous sequence of bits of average length $(1-q)/q^2=9900$ *bits*. Typically, error probability introduces random errors, while burst probability imposes the appearance of continuous errors just after a random error appeared.

In scheme 2, errors result in packet dropping. Typical packet sizes used in our experimentation were inside the range of 100 to 2000 bytes, and packet dropping probabilities with typical values of 10^{-6}, 10^{-5}, 10^{-4}.

The overall simulation is as follows:

- The Kakadu JPEG2000 encoder encodes an image with eight (8) quality layers with the higher layer representing lossless reconstruction, using reversible filters and error resilience capabilities (extra markers in the codestream).
- In scheme 1, the codestream, is corrupted by a noisy channel simulation software, which leaves a variable number of quality layer (the lower 1 to 7) intact and affects on the rest with two error schemes and with various random error and burst error rates. In scheme 2, the codestream is corrupted by a congested channel simulation program, where the hit packets are dropped.
- The Kakadu JPEG2000 decoder decodes the corrupted codestream taking into account the error resilience markers, and reconstructs the image.
- An evaluator (program imgcmp, which is part of the public available jasper system [10]) is used to estimate the resulting image quality.

Results for scheme 1 are shown and interpreted in the following sections. Scheme 2 results could not be included due to paper length limitations, but are available for discussion.

3. ERROR MODELS AND JPEG2000

In this work we have imposed two error schemes – the first one accommodated with two different error models – on a virtual communication channel and simulated the coding and transmission. The two models of error scheme 1 work on bit basis and their impact on a codestream should be, respectively:

0. bit values are flipped

1. bits are dropped (do not reach decoder)

For this error scheme, we run 100 tests on each of the test images, for the two models (0 and 1), three error rates, seven burst factors and eight quality layers.

As mentioned earlier, results produced for scheme 2 are not included in this paper due to paper length limitations.

Results showed that model 1 has the maximum impact on the codestream, from an ‘error rate-distortion’ point of view, regardless of the error probability. Figure 1 shows the average data corruption percentage for the three error probabilities and for the two error models. Values are averaged over all burst probabilities. The effect of burstness is evident in these measurements, since, for example when error rate is 10^{-5} the average corruption becomes about 30%, which corresponds to an effective error rate of $3x10^{-1}$, much higher than the original 10^{-5}.

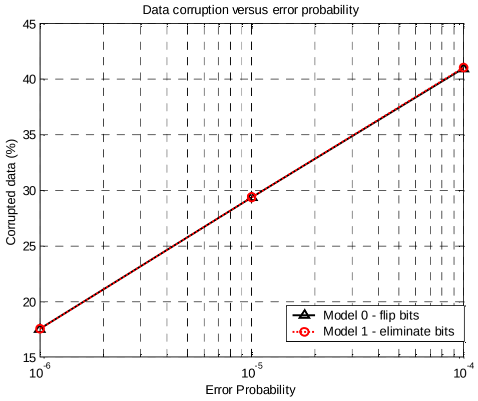


Figure 1. Average data corruption (%) versus error probability for the three error models

One of the interesting outcomes is depicted in figure 2, where decoded image quality is shown versus the percentage of data corruption. One can observe that quality drops rapidly, for example in the 10^{-6} curve for model 0, to about 38dB for data corruption of up to about 10% and remains almost constant (about 3 dB total decrease) for data corruption up to almost 90%. So, when applying model 0, the outcome is proportional to the error and burst error rate present in the network. When applying error model 1, even when corruption is low, the decoded image quality is poor.

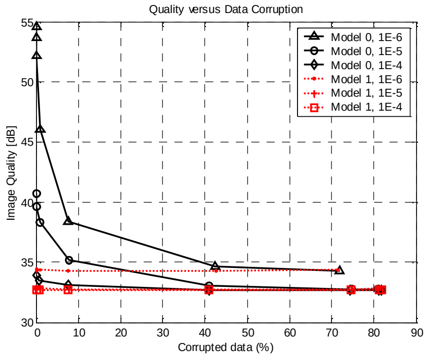


Figure 2. Quality versus data corruption (%)

Another important aspect is illustrated in figure 3, where we show the average quality per number of error-free transmitted quality layers. All curves can be approximated by straight lines with about 2.49 dB/layer slope and can be expressed as $y = 2.49x + b$, where b takes a value depending on the curve (for 10^{-5} model 0, b is 26.35). By knowing the decoding quality for the lower quality layer for a given error model and error rate, we are able to predict the quality of the decoded image whenever additional quality layers reach the decoder free of errors.

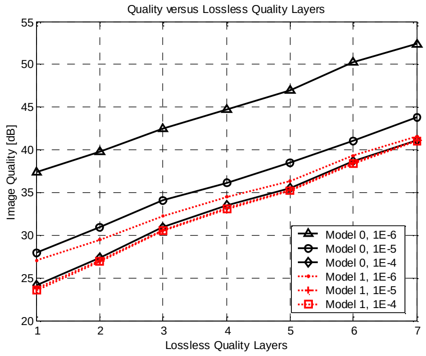


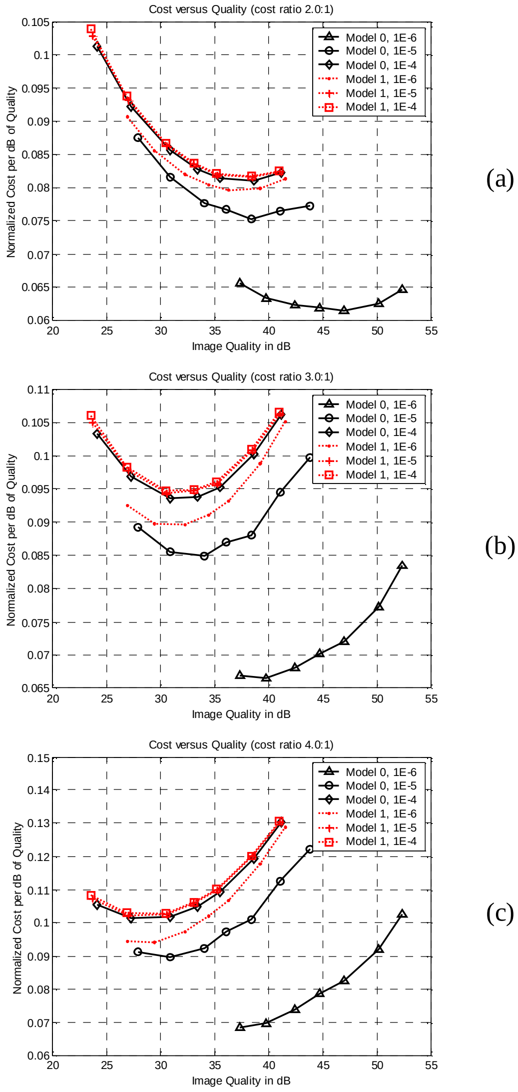
Figure 3. Average Quality versus the number of error-free quality layers

4. COST ANALYSIS

For the purposes of our work, we adopted the following network policy:

- The codestream is divided into droppable and non-droppable parts, in the sense that the network is able to guarantee error-free transmission of the lower quality layers 1 (to 7) (which are considered to be non-droppable or essential). The remaining quality layers (7 to 1 accordingly) are subject to channel errors and are considered to be droppable or additional.
- The non-droppable and droppable data are assigned a cost ratio. Cost ratios tested were from 1:1 to 10:1 in 1:10 increments (91 ratios):

*Cost=non_droppable_data + droppable_data*cost_ratio*



(a)

(b)

(c)

Figure 4. Average cost per dB of quality versus the image quality for cost ratios 2:1, 3:1 and 4:1

Figure 4, depicts the average cost per dB of quality versus the decoded image quality for cost ratios 2:1 (a) 3:1 (b) and 4:1 (c). The data points in these curves are obtained by increasing the number of non-droppable layers from 1 to 7 (0.1 bpp to 2 bpp). From this diagram, one is able to evaluate when the cost of every dB of quality is worth the achieved quality and by how many error-free quality layers can this be accomplished. For example, with 3:1 cost ratio, model 0 and error rate 10^{-6}, the cheapest dB of quality is at about 40dB when using 2 error-free quality layers. So, with a 3:1 cost ratio we must transmit the image up to 0.25 bpp in error-free mode and allow errors to occur to the rest. In our experimentations, cost ratios ranging from 2:1 to 5:1 has only shown to give interesting results concerning policy choices, since within this range the cheapest quality dBs are not located at the edges of the curves.

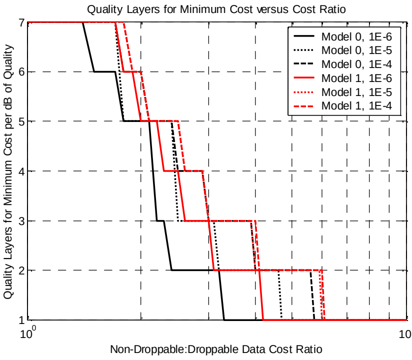


Figure 5. Quality layers for minimum cost per dB of quality versus the cost ratio

In addition to figure 4, figure 5, which is a plot of the number of error-free quality layers adequate to achieve minimum cost versus the cost ratio, makes clear how many quality layers should be transmitted error-free in order to achieve the cheapest quality for every cost ratio. For cost ratio 3:1 (value 3 in x-axis) and for model 0 and error rate 10^{-6}, the number of quality layers is 2. Figures 4 and 5 can provide with the best choice in fees policy (cost ratio) for a given situation of error model, error rate and quality level.

5. CONCLUSIONS

In this work we simulated a communication network assigned with the task to transmit progressive-by-quality error resilient JPEG2000 codestreams through a noisy or congested channel. Extensive tests run on the standard test images using various error schemes, models, rates, burst rates, packet sizes and a variable number of error free transmitted quality layers, gave interesting results concerning not only the error resilience capabilities and restrictions of the JPEG2000 coder but also the cost analysis aspect of such a communication system. For the purposes of this paper, we presented the bit error results and, summarizing, we are able to say that:

1. in a noisy channel (bit-flipping environment),
    - corruption of data and quality of decoded image is proportional to the error and burst error rate. Burstness plays an important role to decoded image quality
    - decoded image quality versus data corruption exhibits a parabolic behavior matching a usual rate-distortion curve
    - data corruption and decoded image quality versus the number of guaranteed transmitted error free quality layers, both exhibit straight line behavior with constant slopes (in the case of quality) or slopes dependent upon the error rate (in the case of corruption)
    - the cost per quality dB exhibits upper and lower bounds in strict relation with the error rate. Careful selection of cost ratios can result in optimum channel operation (from a cost aspect)
    - error-free quality layers vs. cost ratio curves can provide with additional insight aiding in the selection of the appropriate guaranteed error-free bandwidths according to network utilization needs and policies
2. in a bit-dropping environment (channel with outage periods) results differentiate from the ones in a bit-flipping environment in that
    - quality of decoded image is much lower
    - decoded image quality versus data corruption exhibits an almost linear behavior
    - quality costs are higher with lower qualities

6. REFERENCES

[1] C. Chamzas, D. L. Duttweiler, “Encoding facsimile images for packet-switched networks”, *IEEE Journal on Selected Area in Communications*, Vol. 7, No. 5, June 1989.

[2] D. Santa-Cruz, T. Ebrahimi, J. Askelof, M. Karsson and C. A. Christopoulos, “JPEG 2000 still image coding versus other standards”, *proc. SPIE, 45th annual meeting, Applications of Digital Image Processing XXIII*, Vol. 4115, pp 446–454, Aug 2000.

[3] D. Santa-Cruz and T. Ebrahimi, “A study of JPEG 2000 still image coding versus other standards”, *proc. X European Signal Processing Conference*, Vol. 2, pp. 673–676, Sept. 2000.

[4] D. Santa-Cruz and T. Ebrahimi, “An analytical study of JPEG 2000 functionalities”, *proc. IEEE ICIP 2000*, Sept. 2000

[5] C. Christopoulos, A. Skodras and T. Ebrahimi, “The JPEG 2000 still image coding system: an overview”, *IEEE trans. On Consumer Electronics*, Vol. 46, No. 4, pp. 1103–1127, Nov. 2000.

[6] M. Boliek, C. Christopoulos and E. Majani (editors), *“JPEG2000 Part I Final Draft International Standard”*, (ISO/IEC FDIS15444-1), ISO/IEC JTC1/SC29/WG1 N1855, Aug. 2000.

[7] J. Liang and R. Talluri, “Tools for robust image and video coding in JPEG2000 and MPEG-4 standards”, *proc. SPIE Visual Communications and Image Processing Conference (VCIP)*, Jan. 1999.

[8] I. Moccagata, S. Sodagar, J. Liang and H. Chen, “Error resilient coding in JPEG2000 and MPEG-4”, *IEEE Journal of Selected Areas in Communications (JSAC)*, Vol. 18, No. 6, pp. 899–914, June 2000.

[9] http://maestro.ee.unsw.edu.au/~taubman/kakadu

[10] http://www.ece.ubc.ca/~madams/jasper