Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Signal Processing 85 (2005) 1827–1844

SIGNAL PROCESSING

www.elsevier.com/locate/sigpro

A multi-segment image coding and transmission scheme

G. Pavlidis[a], A. Tsompanopoulos[a], N. Papamarkos[b], C. Chamzas[b,*]

[a]*Cultural & Educational Technology Institute, 58 Tsimiski Str., Xanthi 67100, Greece*
[b]*Democritus University of Thrace, 12 Vas. Sofias Str., Xanthi 67100, Greece*

Received 30 April 2004; received in revised form 11 April 2005

Abstract

In this paper, we present a multi-segment image coding scheme suitable for lossless progressive transmission over noisy or congested networks. The method is a block-wise forward-adaptive context selection technique that employs a low-complexity classification process to separate an image into multiple segments according to local statistics, so that compression efficiency by an entropy coder is improved and progressive transmission becomes possible. At the same time, loss of information in case of error occurrence during transmission results in less obvious quality degradation as the lost blocks are scattered in the final image. The scheme was extensively tested and turned out to give results comparable to those given by state-of-the-art standards, such as L-JPEG, JPEG-LS, CALIC and FELICS, in terms of compression efficiency, while providing with scalability and error concealment mechanisms.

Keywords: Image coding; Image transmission; Block-based image segmentation

1. Introduction

Compression is the coding of data to minimize its representation either by removing redundancy or by removing information less perceivable by the human observer. Efficient *lossless* compression of images can be made possible by modeling the data using non-uniform distributions. There are many applications that require lossless image compression such as in medical imaging, image archival and remote sensing [1-10]. State-of-the-art lossless compression schemes, such as JPEG-LS [6,7], Context-based Adaptive Lossless Image Coder (CALIC) [8] or Fast and Efficient Lossless Image Compression System (FELICS) [9], were built upon certain sophisticated assumptions about image data to be encountered. The main idea in such schemes is to divide encoding into two independent phases: modeling and coding. At the modeling phase, the image samples are observed at a predefined order and information about the data is gathered to form a probabilistic model that best describes the image. At the coding

*Corresponding author. Tel.: +30 25410 79571, fax: +30-25410-63656
E-mail address: chamzas@ee.duth.gr (C. Chamzas).
URL: http://www/ceti.gr, http://www/duth.gr, http://www/duth.gr.

0165-1684/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.sigpro.2005.04.006

phase, code assignment takes place in a sequential (adaptive) or in a two-pass (learning) manner. At that phase, the error generated by subtracting the predicted data values from the actual ones is encoded. Another equivalent approach to lossless image compression is to employ a *lossy+lossless* compression scheme, where:

- the original image is encoded with lossy compression, and
- the residual (or differential, or error) image that is produced by the subtraction of the encoded image from the original is losslessly compressed in order to be able to reconstruct perfectly the original image.

Of course, there are various compression algorithms available (DCT or DWT-based) that can provide with the needed separation into a lossy-compressed and a residual image.

The residual image, as defined, has, typically, a largely reduced variance, compared to the original image, and significantly less correlated pixel values. Fig. 1 shows the probability distributions of the gray-level image "Lena", the (lossy) JPEG compressed "Lena" at 0.25 bpp and the residual image produced by subtracting these two images. Since the residual image is the difference of two 8-bit quantities it is represented by 9 bits (i.e. values ranging from −255 to 255). As shown in Fig. 1(c), the probability distribution of the residual is highly peaked around zero and can be closely approximated by a double-sided exponential distribution of the form:

$$p = \frac{1}{\sqrt{2\sigma^2}} e^{-\sqrt{\frac{2}{\sigma^2}}|x|} \tag{1}$$

also known as the Laplacian distribution or the two-sided geometric distribution, with zero mean and variance σ^2. It is long accepted that such residual images have roughly the same shape of distribution, and differ only in their variance σ^2 [11–14]. It has also been proven that the most appropriate methods to encode such images incorporate entropy coders.

In this work, we propose a method that:

- uses a standard lossy compression scheme to produce a lossy and a residual image,

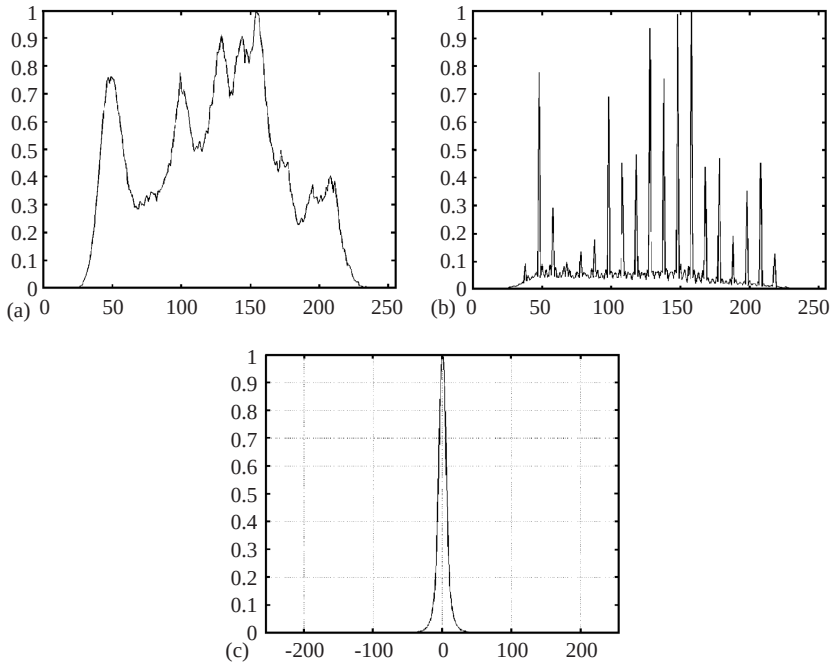

Fig. 1. Probability distributions of gray-level image "Lena": (a) original image, (b) compressed image, and (c) residual image.

- uses a simple classification scheme to separate the residual image into N arbitrarily shaped segments, and
- encodes, separately and individually, the N segments of the residual image using entropy coding.

Fig. 2 depicts an overview of the proposed system. The system is suitable for progressive transmission and reconstruction of the encoded residual segments. Well-known algorithms such as the DCT [16] and the DWT [17-21] were tested as being the lossy codec, through the application either of JPEG [16] or JPEG2000 [22-26], respectively, and an arithmetic coder [27] was employed to produce the final codestream. Our approach in segmenting the residual images was based on employing the Kohonen's Self Organizing Feature Map (SOFM) neural network [28-32], mainly because of its simplicity and effectiveness in such classification tasks.

The main contribution of this work is the specific set-up that is proposed, which is based on using existing standards and methodologies to build a system that supports efficient compression and progressive transmission. The benefits of such a system are multiple:

- By applying segmentation to a residual image using criteria matching an entropy encoder we are guaranteed to achieve efficient compression.
- By encoding the residual segments separately, we are able to achieve progressive transmission and reconstruction in the decoder (scalability).
- By transmitting the residual segments sequentially though a noisy or congested network, we are able to provide with some sort of an indirect error concealment mechanism.

This set-up describes, actually, a generic forward-adaptive context selection technique, since there is a preprocessing step (segmentation and classification) prior to the entropy coding, which provides with data for the modeling that have to be transmitted to the decoder to guarantee a proper decoding.

The method was tested on standard gray-level test images and resulted in average bit-rates comparable to the ones obtained by using standard lossless JPEG, JPEG-LS [6,7] and JPEG2000. The advantage, as stated, over such schemes (except of course JPEG2000) is that the proposed method supports progressive lossless reconstruction of the original image by partitioning the residual data into classes and rearranging the transmitted codestream so that quality degradation through packet loss is less conceivable and obvious, as errors are scattered into the final decoded image, and all this through a low complexity segmentation process.

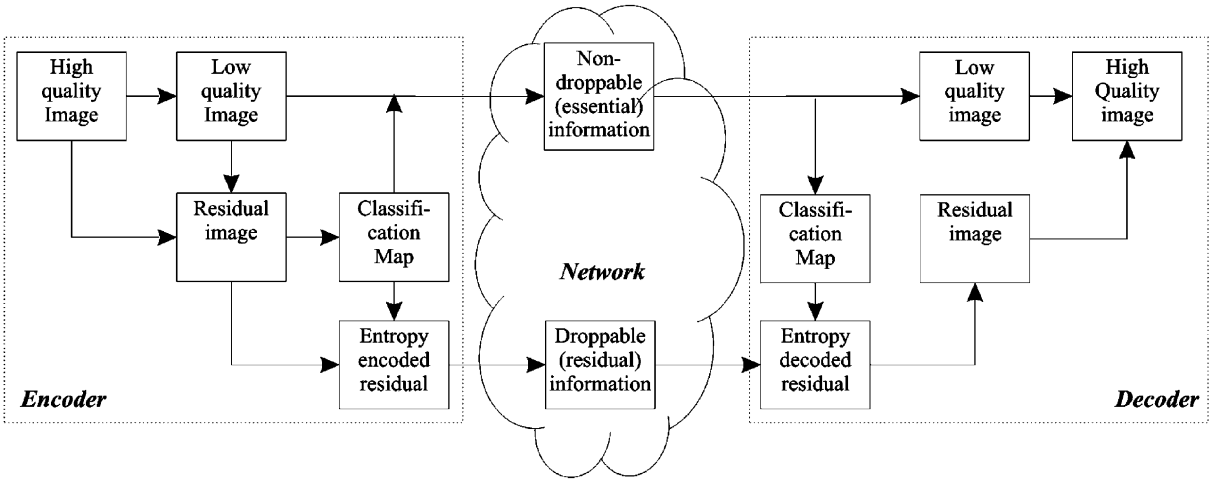

Fig. 2. Block diagram of the proposed image coding scheme.

2. The multi-segment lossless coding scheme

The Multi-Segment Lossless Encoder (MSL-E) (Fig. 3) consists of two basic coding stages: the lossy encoder (which can be DCT or DWT-based followed by quantization) and the residual data classifier with the entropy coder: an original image is compressed through a lossy encoder, and the compressed image is transmitted as the low-quality replica of the original. The same compressed image passes through a decoder and the decoded image is subtracted from the original to produce a residual image. The residual image is fed to a neural network classifier to obtain a classification map based on image tile characteristics (tiles, hereafter, refer to non-overlapping rectangular blocks covering the image). The tiling is imposed in order to capture local characteristics and to be able to classify the image based on these local features, thus forming a forward-adaptive context selection mechanism for the arithmetic encoder at the final stage. Both the low-quality image and the classification map are considered to be the "essential" or the "non-droppable" parts of the content.

Typically, residual images consist of large "flat" areas with zero mean and small variance, and of areas with "edges" or "activity" where variance grows larger and distributions are highly skewed. The effect of the lossy transform coding (at the first stage of the system), due to its energy compaction nature and the fact that images are treated as noisy signals, is to degrade edge information. As a result, information about the edges is expected to reappear in the residual image.

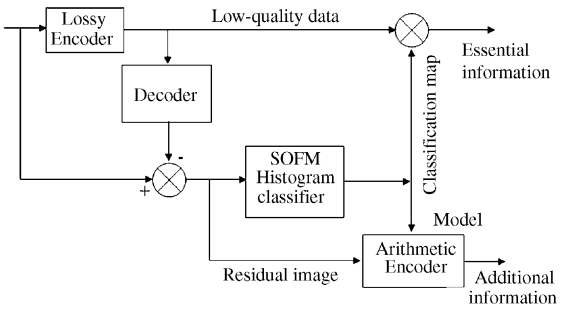

Fig. 3. The Multi Segment Lossless Encoder (MSL-E).

Considering that these edges are usually of small width (after all they represent rapid changes in images), one can expect to be able to capture them with high probability inside small blocks of the image. That is why the image is tiled into small blocks and is classified on that basis. The obtained classification map is used to segment the residual image and to provide the arithmetic encoder with contextual information, forming a prediction model.

2.1. The lossy encoder

Two encoders were tested as being the first coding stage of the proposed system: JPEG[1] and JPEG2000.[2] When using JPEG, coding is controlled by a parameter called "quality factor" ranging from 1 to 99, which is directly related with the imposed quantization. When using JPEG2000, coding can be controlled by directly providing a target bit-rate.

When considering the case of transmitting through noisy communication networks, low-to-medium bit-rates would be preferable for the "essential" part: the idea is to send as little as possible essential data over the channel, using a guaranteed lossless transmission, and to progressively enhance quality using the additional residual segments, that are sent through lossy transmission. Typical bit-rate selections ranging from 0.4 to 0.7 bpp (for the essential data) gave the best overall average bit-rate results when using JPEG. On the other hand, the usage of JPEG2000 can provide with acceptable quality and similar overall performance by targeting even lower bit rates for the essential data.

2.2. The block classifier

In vector quantization (VQ) [33,34], an image is represented by k-dimensional vectors. Every such

[1]We have used the standard JPEG coder cjpeg v.6b, available by the Independent JPEG Group.We have used the standard JPEG coder cjpeg v.6b, available by the Independent JPEG Group.

[2]We have used the Kakadu [39] system for JPEG2000 coding, developed by David Taubman.We have used the Kakadu system for JPEG2000 coding, developed by David Taubman.

vector x is compared with "model" vectors (the so-called code-vectors that form a codebook), which are usually the product of a learning process. This way, VQ establishes a mapping of image vectors onto pointers to code-vectors. VQ is known to be a very efficient way to quantize image data but is also known to be a very computationally demanding process. Under the scope of reducing the complexity involved in a typical VQ, an alternative would be to implement a quantizer using neural networks (NNs): NNs can be used both to produce code-vectors (during a training process) and to quantize images accordingly. *This approach is equivalent to the implementation of a fast VQ with reduced complexity*.

In the proposed system, an Unsupervised Neural Network Classifier (UNNC) was employed to classify the residual image tiles, by creating a mapping of each image tile to an appropriate class. This UNNC was the Kohonen's Self-Organizing Feature Map (SOFM), an NN algorithm that is frequently used as unsupervised classifier on gray-level images [35]. This classifier is, hereafter, referred to as VQ-NN. The VQ-NN can operate either in an *adaptive* or in a *static* mode: this adaptivity refers to the process of training, and is the case when the codebook is being generated by sampled data from the image to be quantized. On the other hand, when representative images are being used to train the network before its actual application to quantization, then a static codebook is being formed. Obviously, the static mode has the speed advantage while the adaptive mode has the accuracy advantage. Nevertheless, the adaptive mode can be simplified and speed-up by imposing some restrictions and limitations.

The proposed classification process is as follows:

- the residual image is divided into tiles of size 8×8 (or bigger),
- the histogram of each tile h is computed and a huge matrix H of all tile histograms is generated,
- the overall histogram matrix H is fed to the SOFM. The matrix can serve both as the training (in the adaptive mode) and the working data set. The SOFM is trained to adapt to the histogram data, and then classifies the image tiles in accordance with their corresponding histograms: *tiles are, therefore, classified by the similarity in the shape of their histograms*,
- the number of output classes was chosen to be eight (8), while there is the option of using more (i.e. sixteen) classes when more and smaller individual image segments are needed.

The Kohonen SOFM employed consists of a simple neuron topology of 511 inputs and 8 outputs. The inputs can assume any real value, whereas outputs are forced to be binary decisions (0 or 1). Note that the inputs are 511 because the histograms have a range of values inside $[-255,255]$ due to the 9-bit nature of the residual images. The input neurons $i = 1{:}511$, reflect the indices of the histogram data presented: input neuron i reads the probability of gray level i.

The SOFM is initially trained by presenting all, or parts of the image tile histograms until a mean squared error (MSE) criterion is satisfied. The process of training is typical and straightforward and is implemented within the NN algorithm. After the training session, the SOFM is ready to classify the input histograms to the selected number of output classes. The NN operates in a winner-take-all mode, where a specific "shape" of input histogram votes for one output neuron, satisfying the model that was developed during the training.

The final outcome of the classifier is a classification map that reflects a mapping of a 9-bit residual image to a 3-bit (for 8 classes) coarse image. This classification map can be used to segment the residual image and feed contextual information to the next coding stage, the arithmetic coder. Fig. 4 shows the average histograms of two distinct classes after the classification (for illustrative purposes the histogram intervals are shown for $[-128,127]$ while the actual interval is still $[-255,255]$).

In Fig. 5 a typical classification map is shown. Different pixel intensities correspond to different classes. This map was obtained by the classification process on the residual image generated by subtracting from the original gray-level image "Lena" the DCT compressed version at 0.25 bpp (this residual image gave the histogram shown in Fig. 1(c)). It is notable here how the classifier

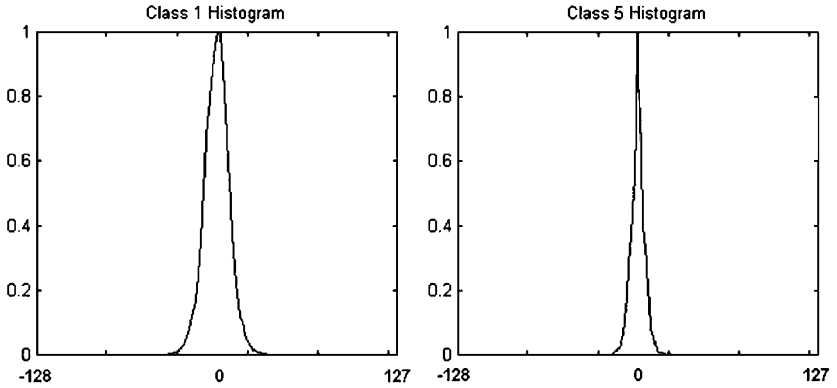

Fig. 4. Class 1 (a) and class 5 (b) histograms.

Fig. 5. A typical classification map obtained by the classification process.

captures and enhances information content of the original image from within the residual image. Image regions with similar characteristics are clustered together.

Constraints can be imposed on the input matrix H in order to reduce its dimensions and the corresponding computational complexity and processing time. As stated before, any residual image exhibits a Laplacian probability distribution with values inside a relatively small interval $[-a, b]$. The same principle applies to individual tiles. In a typical situation where compression of about 0.25 bpp is selected at the initial stage of lossy encoding, the residual image histogram falls inside the interval $[-50,50]$ (on the average). This histogram can be adequately described by the values inside this interval, and consequently, values outside this interval may be clipped out. This clipping results in reducing the number of input neurons needed to describe the data (to 101 in this case), which, in effect, reduces dramatically both the training and classification times. The results shown in this work were obtained by clipping-out zero values at the boundaries of the histograms. In order to reduce even more the computational load, additional constraints to the training procedure could be imposed: the SOFM could be trained once, by presenting many sample histograms matching the actual data, and expect it to perform well in the future without any more training (since all residual images exhibit similar probability distributions). Fixing the NN to predefined states can be thought of as a *static operational mode*, resulting in significantly reduced encoding times, as opposed to the *adaptive operational mode*, where the NN is trained for every image. The trade-off between classification accuracy and computational time should be well balanced for a specific kind of application, since the training procedure of a SOFM consumes a significant amount of the total encoding time.

2.3. The arithmetic encoder

A typical arithmetic coding (AC) system consists of two operational stages: the *engine*, where we feed symbols and their probabilities, and a *model* that is responsible for the probability estimation of the current symbol. One can think of the model as the "intelligence" of a compression scheme. The model is responsible for deducing or interpolating the structure of the input, whereas the coder is the "engine room", which converts a probability distribution and a single symbol drawn from that distribution into a code. A good model will produce symbol probabilities near 0 or 1. One of the most important features of the arithmetic coding is that the engine does not depend upon any particular model, while providing optimal coding for any given model.

The final stage of the proposed encoding system is such an arithmetic encoder. The arithmetic encoder uses contextual information from the classification map and the low quality image and encodes accordingly the residual image. The output can be either one encoded codestream, or as many codestreams as the classes found in the classification map, and, thus, transmitted as one or separate codestreams respectively. Every segment represents the encoded part of the residual image that belongs to a particular class. The main advantage of using segments, especially when transmitting over noisy communication channels, is that the loss or drop of a packet from any segment will result in loss of residual information that belongs only to that class. Moreover, since each class is distributed over the entire image, *the effects of blocking artifacts, especially at the boundaries of blocks that belong to different classes, are indirectly reduced and the final image quality degradation is less obvious, as errors are scattered throughout the image surface*.

2.4. Transmission over congested networks

The emerging need for multimedia transmission over high-speed communication systems, on which several restrictions and transmission policies apply, guide the research to new ways of data handling and algorithm applications, even before the data can be transmitted. One of the most basic 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 [15].

Standard compression methods, such as JPEG or JPEG-LS, provide with efficient compression, but, unfortunately, they 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.

Several tests considering this aspect have been done for the proposed system. These tests aim at proving the ability of the system to provide with indirect error concealment, as well as outlining the possibility to impose flexible cost policies. It has to be noted that the simulations address the case of a purely theoretical network facility that provides with no error resilience mechanisms and suffers from both bit errors and packet dropping.

3. Experimental results

The system that was developed (and implemented in MatLab and in ANSI C) was used in a series of different experimentations in order not only to understand and analyze its behavior but also to validate its theoretically assumed advantages. To this end, several tests have been designed to:

- confirm the assumed exploitation of the imposed segmentation by the arithmetic coder,

- confirm the superiority of the imposed segmentation over a simple tiling in the overall coding efficiency,
- test several prediction models in the arithmetic coder for compression optimization,
- test several compression methods and parameters for the generation of the low quality image (which, of course, leads to the generation of a different corresponding residual image),
- test the usage of different numbers of target classes during the residual image segmentation, under the scope of improving compression efficiency and aiding controllable network transmission,
- test the system's efficiency during lossy transmission over channels with various error rates, and
- confirm the possibility of using the scheme in real-life communication networks due to its inherent property of being able to provide with flexible cost control mechanisms that can be adjusted to specific user requirements and provider policies.

All these tests where carried out using several gray-level images that include real-world photos, computer generated graphics, low quality web-cam images and textual images [36]. Before presenting the results of these experiments, an overall system description is provided here, through the usage of an example. Images produced during the process are shown in Fig. 6: (a) shows the original test image "Lena" (8 bpp 512×512 pixels); (b) shows the low quality image generated by applying JPEG compression to the original image (image is compressed at 0.33 bpp); (c) shows the residual image and (d) is the classification map produced by the application of the VQ-NN on the residual. For the generation of the low quality image by JPEG compression (in this example and all the experiments) the standard JPEG coder *cjpeg v.6b* provided by the Independent JPEG Group [37], which is included in the Cygnus distribution [38] was used. The quality of this low-quality image was calculated to 32.95 dBs PSNR. The generated residual image had a dynamic range inside $[-55, 57]$. It was segmented to eight (8) classes using a tiling of 8×8 blocks. Due to this fact, the classification map was of 3 bpp and 8 times smaller than the original image in both axes. It was encoded using lossless compression using a typical arithmetic encoder. Finally, the generated residual segments were encoded using adaptive arithmetic encoding, leading to an overall data bit-rate of 4.35 bpp. It should be noted that for illustration purposes the gray-levels of the images in (c) and (d) were shifted and scaled using contrast stretching.

3.1. Confirmation of the assumed exploitation of the imposed segmentation by the arithmetic coder

As known by many works that have analyzed the arithmetic coding, the efficiency of the arithmetic encoding is significantly based upon the type of the probability distribution of the input data. For example, an input with uniform distribution will not be compressed at all by an AC. On the other hand, distributions with low variance and less uniformity are preferable. On a basis of higher order moments, such as skewness and kurtosis, higher values are preferable.

Based on these theoretical facts, the confirmation of the assumed exploitation of the imposed segmentation by an AC to improve compression efficiency was based on a simple experiment: the typical gray-level test image "Lena" was compressed with JPEG for the generation of the low-quality image. The JPEG compression was done using all quality factors from 5 (0.17 bpp) to 95 (2.8 bpp), thus leading to the generation of many low-quality images. These low-quality images were used to produce the corresponding residuals that were segmented and classified using the histogram classifier to 8 classes on an 8×8 pixel basis. Finally, the residual segments were compressed by AC. The overall average values of moments of the whole residual image as well as these of the segments are shown in Table 1 for the full scale of low-quality-image bit-rates ranging from 0.17 to 2.8 bpp. As shown, the average value of variance in the residual segments is much lower than the one of the whole residual image. Additionally, the higher order moments (skewness and kurtosis) demonstrate a significant increase in their average absolute values. These results show that the imposed segmentation and classification lead to

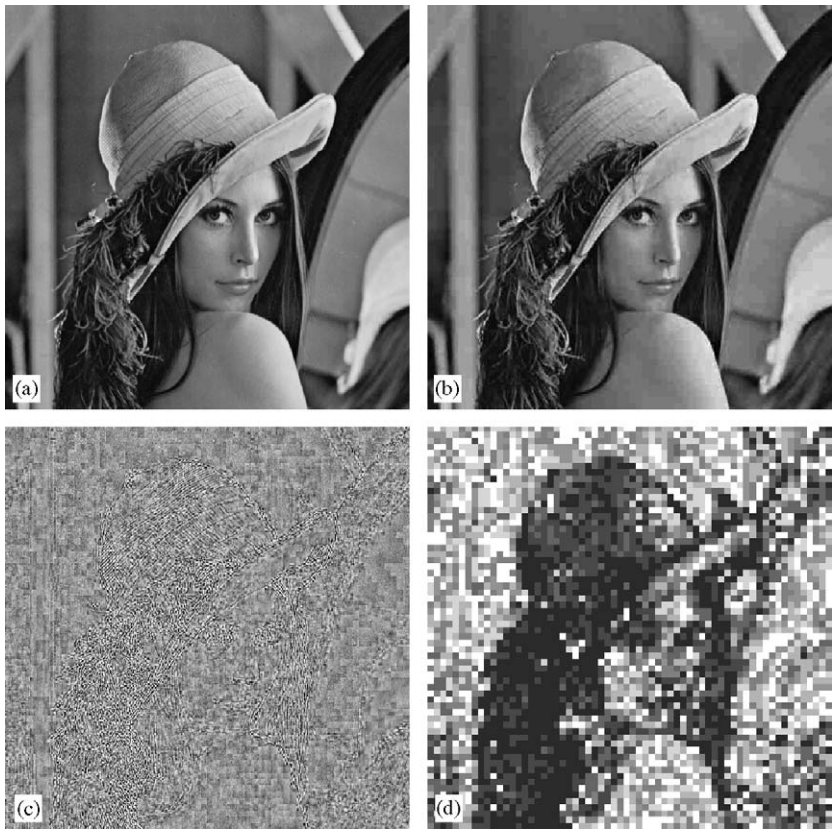

Fig. 6. (a) Original image of 8 bpp, (b) image after JPEG compression up to 0.33 bpp with estimated quality 32.95 dB PSNR, (c) corresponding residual image, and (d) classification map (8 classes, 8×8 blocks, magnified to original image size).

Table 1
Average values of moments in the whole residual versus the values in the residual segments

	Variance	Skewness	Kurtosis
Whole residual image	25.051	0.054	2.817
Residual segments (average)	5.528	0.674	48.960

the formation of the residual data in such a way that favor the usage of an AC, confirming the assumed possibility of improving adaptivity of data to the AC engine.

3.2. Confirmation of the superiority of the imposed segmentation over a simple tiling in the overall coding efficiency

For the confirmation of the superiority of the imposed histogram segmentation and classification of the residual data over a simple tiling in the provided data adaptation to the AC and, thus, the overall compression efficiency, a simple experiment was conducted: after the generation of the residual image, two segmentation methods were imposed, either the proposed histogram classification or a simple sequential tiling. The rest of the system was left intact. Comparative compression results are shown in Table 2. The gray-level image "Lena" was again used in this experiment, having been compressed up to 0.33 bpp as a low-quality image. The results show that compression efficiency, when using histogram segmentation, was improved by a 4%.

Fig. 7 shows a graphic comparison of the average performance (over all test images and encoding templates) of the proposed scheme compared with the simple tiling scheme. The curves in this graph represent the achieved

Table 2
Segmentation versus tiling within the core of the proposed system

(size in bytes)	Segmentation	Tiling
Low-quality image	10 860	10 860
Compressed classification map	770	—
Compressed residual image	142 386	148 614
Total size [lossless]	154 016	159 474
Overall bit-rate (bpp)	4.70	4.87

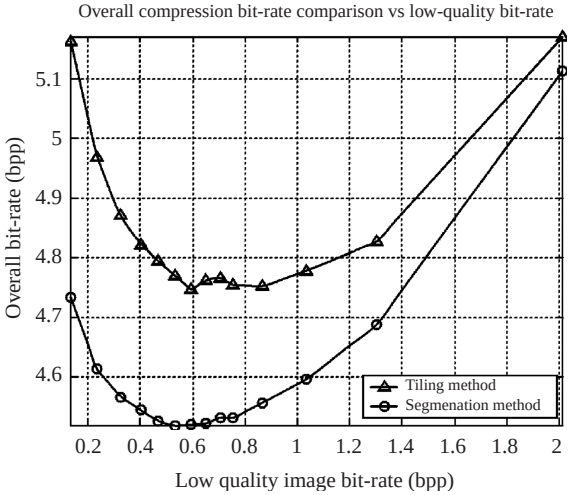

Fig. 7. Average overall compression bit-rate versus the low-quality image bit-rate when using segmentation or tiling.

bit-rate as a function of the bit-rate of the low-quality image. It is apparent that the proposed method always outperforms a simple tiling in terms of the achieved lower overall average bit-rate. Additionally, from this figure it can be deduced that best compression is achieved when the low quality image is compressed to approximately 0.6 bpp.

3.3. Testing several different prediction models for compression optimization

As stated in previous paragraphs, the modeling of a source plays the most important role when this source is to be encoded with AC. In order to improve compression efficiency several prediction models and context templates were tested. Some tested prediction models and templates are shown in Table 3, where the topology of pixels is as shown below (with letter "x" representing current pixel, while letters "a" to "h" represent the immediate neighbors):

Table 3
Segmentation versus tiling within the core of the proposed system

Model	MSB 12 bit Template LSB											
	11	10	9	8	7	6	5	4	3	2	1	0
1					g	c	e	a	x	x	x	x
2	g	g	c	c	e	e	a	a	x	x	x	x
3	c	c	c	c	a	a	a	a	x	x	x	x
4					g	c	e	a	x	x	x	1
5	c	c	c	c	a	a	a	a	x	x	x	0
6									x	x	x	0
7-1					g	c	e	a	x	x	x	1
7-2	c	c	c	c	a	a	a	a	x	x	x	0

b	c	d
a	**x**	e
h	g	f

These models take into account either the classification map, either the low-quality image, either both, forming various combinations, by using several context templates. These context templates are of 12 bits at-most (Table 4 shows an example model). This implies the existence of $2^{12} = 4096$ contexts. Such a number of contexts is relatively large for the implementation of a fast-adapting AC. Experiments confirmed this observation by demonstrating better compression bit-rates when using models with small contexts. Additional study of the images to be used and manipulated encouraged the idea of imposing a conditional model, such as model 7, which switches between two other models (namely 4 and 5), where significant changes in the probability distributions are expected. Such situations are at tile boundaries, where for the upper and left edge pixels model 4 performs well, whereas

model 5 performs efficiently in all other pixels inside the tile.

Overall average comparative results of employing several such prediction models (over the whole set of test images) are reported in Table 5 in terms of achieved overall average bit-rates.

3.4. Testing several compression methods and parameters for the generation of the low-quality image

Due to the initial idea and design of the proposed method to be based upon generic coding techniques, there is not a direct relevance of the efficiency of the scheme with any specific coding method that will be employed for the generation of the low-quality image. This irrelevance does not refer mainly to the final outcome of the scheme (in terms of compression or quality efficiency), but to the freedom it provides in choosing any method appropriate to specific needs, limitations and requirements. The main contribution of the proposed method is in its ability to handle efficiently the residual data that are produced in the process. And, as already mentioned, these residual images are always of similar statistics no matter what the encoder is. From this point of view, the low-quality image does not play a very important role on the efficiency of the system.

To confirm these thoughts, comparative tests have been designed and conducted that involved the usage of JPEG and JPEG2000 for the generation of many low-quality images compressed to a wide range of bit-rates. The results showed that the usage of JPEG2000 leads to an overall improvement in the compression efficiency and the estimated image quality, but the complexity involved is much higher (due to the nature of the specific method). Table 6 summarizes these overage average compression results. As shown, despite the fact that the total size of the significant information (low-quality image + classification map) is almost identical in both cases, the final overall result favors the usage of JPEG2000. During these tests, no computation times where measured and this is why they are not included in this Table. Of course, the advantage of JPEG in higher speed and lower computational load is apparent.

Table 4
Example of a Model: The case of AC Model 3

MSB					12 bit Template						LSB
11	10	9	8	7	6	5	4	3	2	1	0
c	c	c	c	a	a	a	a	x	x	x	x

A total of 12 bits are used. The Template is filled with bits from the classification map, starting with the LSB:
4 bits from the current pixel (the 4 most significant bits of the pixel value),
4 bits from the pixel to the left (the 4 most significant bits of the pixel value),
4 bits from the pixel above (the 4 most significant bits of the pixel value).

Table 5
Average overall bit-rates for seven (7) prediction models

Prediction models						
1	2	3	4	5	6	7
4.54	4.63	4.52	4.63	4.52	4.63	4.50

3.5. Testing the usage of different numbers of target classes during segmentation

Under the scope of choosing the right number of target classes to map a residual image, in order to improve compression and transmission efficiency, tests were conducted using two candidate numbers: eight (8) and sixteen (16). The

Table 6
Comparative results from using two different encoders to produce the low-quality image

(bpp)	Low-quality image bit-rate	Classification map bit-rate	Significant data bit-rate	Overall data bit-rate
JPEG	0.40	0.01	0.41	4.02
JPEG2000	0.41	0.01	0.42	3.97

decision for these numbers was based on the fact that they are both powers of two and, therefore, the produced classification map would be a paletted image with an integer bit-depth (i.e. 3 or 4 bpp). From these experiments, the final outcome was that there is no significance difference in the overall average efficiency. What differentiates the choices lies in the better control over the progressive transmission that is provided by the choice of using more classes, which implies the transmission of more and smaller residual segments.

Additionally, other choices for the size of the tiles to segment the image have been investigated. Namely, the tests included 8×8 and 16×16 pixel tilings. The initial choice of 8×8 pixels was, actually, guided by the fact that JPEG was used for the generation of the low-quality image. The idea was to use the same tiling as JPEG in order to preserve some sort of "compatibility" with that standard. The overall average results favor this choice of tiling, mainly because the VQ-NN classifier gave better classification results, leading to better prediction models and, finally, more efficient arithmetic encoding.

3.6. Testing the system's efficiency during lossy transmission over channels with various error rates

The final phase of tests involved noisy network simulations. Specifically, a theoretical network was simulated, which did not include any error resilience mechanisms, which exhibits bit and packet errors of various rates. The main concern during these tests was to simulate a congested network. Significant contribution of the proposed system in such situations is in its nature to segment the information to non-droppable and droppable. The system provides this possibility by transmitting the non-droppable information at first, and then transmitting the droppable information in sequential homogeneous parts. Thus, a possible application scenario for the system over a noisy network, from the point of view of a service provider, would include:

- the provider should be able to guarantee the safe (lossless) transmission of the non-droppable (significant) packets under a specific cost policy, and
- the provider could give the user choices for the cost policy imposed to the droppable (supplementary) packets, depending on the required bandwidth and expected error rate.

Again, the experiments in this case were comparative, using either histogram segmentation or simple tiling. In these tests the image blocks were of 8×8 pixels and the target classes were eight (8). After the encoding, the encoded information was transmitted through a virtual network, which imposed errors to the droppable part of the stream with error probabilities varying between 10^{-6} and 10^{-1}. Finally, the parts of the transmitted images that reached the receiver were decoded and the images were reconstructed and compared with the originals. It has to be noted that these experiments simulate the congestion conditions by adopting a worst case scenario where the occurrence of an error results in losing the rest of the segment being transmitted due to loss of synchronization. This test scenario complies with simulations in our previous work on JPEG2000 that tested the error resilience capabilities of the standard and provided with a cost analysis [40,41].

Fig. 8 shows an example of the effect of a congested network on the codestream produced by the proposed coding scheme. For comparison reasons, the effects of the same congestion conditions on the tiling scheme are also depicted. The original image is compressed with JPEG using quality factor 15 at 0.27 bpp. The produced residual image is segmented and encoded by the proposed classification scheme (using 8×8 region histograms). It is also segmented by a simple rectangular tiling method. It is then hit by random noise. Fig. 8(a) shows the residual image after it has been segmented, hit by errors and reconstructed in the decoder side. Fig. 8(b) shows the reconstructed residual image transmitted in simple rectangular tiles. The lost parts are drawn in white to denote the missing information. Fig. 8(c) and (d) show the final reconstructed images and the associated mean squared error and PSNR measurements. As obvious from these images, the distortion imposed in the case of using the histogram classifier are scattered all over the image

Fig. 8. The effect of a congested network on the reconstructed image: (a) reconstructed MSL-coded residual image, (b) reconstructed tiled & JPEG-coded residual image, (c) final reconstructed MSL-coded image and, (d) final reconstructed tiled & JPEG-coded image.

and are much less observable than in the other case.

Fig. 9 shows a quality comparison graph, where the average final decoded image quality is depicted as a function of the low-quality image bit-rate for a cost ratio value of 0.5. As shown in this graph, the histogram segmentation leads to better results compared to a simple tiling, over the whole range of low-quality image bit-rates.

Table 7 summarizes the overall quality results. These results show that the advantage of using the histogram classifier is not so significant when viewed from an objective-quality point of view. The advantage is gained in terms of subjective image quality through the scattering of the errors over the image surface. This subjective quality advantage was confirmed by typical subjective tests with the kind cooperation of our fellow

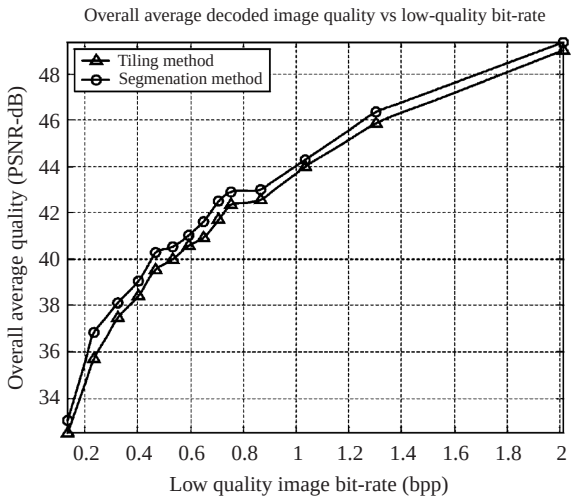

Fig. 9. Average final decoded image quality versus the low-quality image bit-rate when using segmentation or tiling.

Table 7
Comparative results for lossy transmission when using segmentation versus tiling

Measured quantity	Segmentation	Tiling
Max difference	112	133
RMSE	4.17	4.67
PSNR (dB)	35.72	34.74

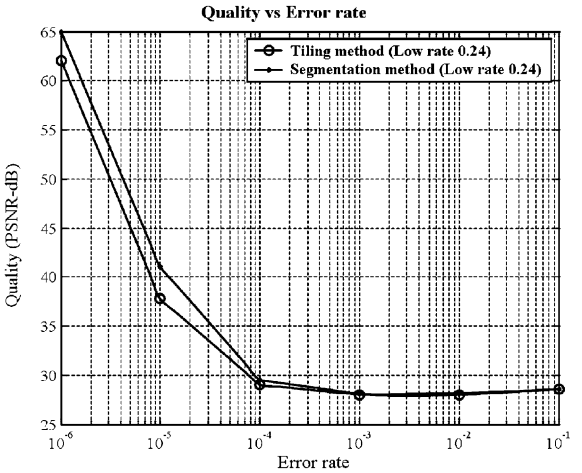

Fig. 10. Overall average compression quality versus the network error rate.

colleagues and engineers from the Democritus University of Thrace, who are not directly related with the subject of image processing.

Fig. 10 shows a graph of the average final image quality as a function of the network error rate. In this experiment, the low-quality image is generated by JPEG compression at 0.24 bpp and the range of network error rates is between 10^{-6} and 10^{-1}. As shown in this graph, for high error rates (10^{-3} and more) imposing or not any classification on the segmented residual image does not make a difference, at least from a quality perspective. Even though, the subjective quality still favors the histogram classification.

Furthermore, for the experimental confirmation of the possibility to impose a cost policy, the following test scenario was adopted:

- non-droppable bytes are assigned a cost value of 1 cost unit per byte of information,
- a cost ratio is defined and droppable bytes are assigned a lower cost value in accordance with this cost ratio,
- the total cost is defined as:

$$\text{Cost} = \text{non\_droppable\_bytes} * 1 + \text{droppable\_bytes} * \text{cost\_ratio [cost units]}, \quad (2)$$

- low-quality images are being created by using JPEG compression with quality factors 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 60, 70, 80, 90 (the total corresponding average non-droppable bitrates would be 0.17, 0.29, 0.40, 0.49, 0.57, 0.64, 0.71, 0.77, 0.84, 0.89, 1.02, 1.21, 1.51, 2.13 bpp),
- costs are normalized: costs of non-droppable and droppable data are normalized by their summation, while overall data costs are normalized by their maximum value.

Fig. 11 demonstrates how the total cost of transmission is affected by the adopted scenario. The three curves of this graph correspond to the cost of non-droppable, droppable and total data. In this experiment the cost ratio was set to 0.5. An important aspect here is that by varying the cost ratio one is able to control the cost of the non-droppable as well as the droppable data while

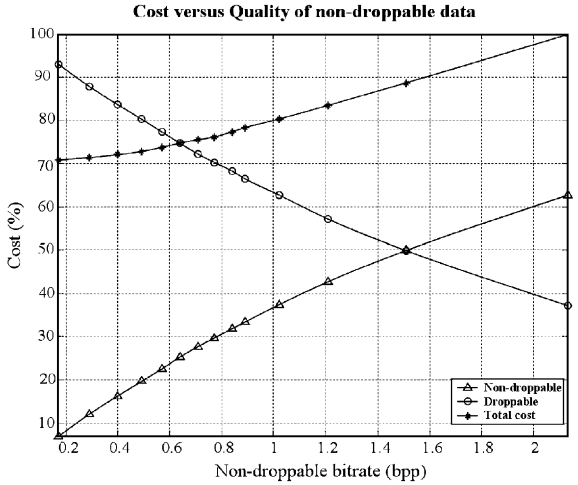

Fig. 11. Overall average cost versus the quality of non-droppable data.

ensuring that some requirement are met according to one of the possible scenarios:

- With cost ratio set to 0.5, the cost is balanced (50–50%) for droppable and non-droppable bits when the non-droppable bit-rate is 1.51 bpp (the position in Fig. 11(a) where non-droppable and droppable curves meet). The overall cost at this point is about 89% of the maximum total cost.
- With cost ratios of less than 0.5, for example 0.1, the cost is 50–50% for droppable and non-droppable bits when the non-droppable bit-rate is 0.4 bpp. The overall cost at this point is about 34% of the maximum total cost.
- With cost ratios of more than 0.5, for example 0.9, the cost is 50–50% for droppable and non-droppable bits when the non-droppable bit-rate is 2.13 bpp. The overall cost at this point almost reaches the maximum (100%) of the total cost.
- It is clear, that if a very low cost ratio is selected (cheap droppable bits), then the average cost of droppable bits is lower than the one of non-droppable almost everywhere in the bit-rate scale (except in very low bit-rates: 0.17, 0.29 bpp). This would be an excellent choice from the customers' point of view.
- If a cost ratio near 1 is selected (expensive droppable bits), then the average cost of droppable bits would always be higher than the cost of non-droppable, and the overall cost would always be near maximum. This would be a very bad choice for a customer, since, whatever the congestion conditions the total cost would always be near maximum.
- If a balanced cost ratio is selected (say 0.5), then the costs of non-droppable and droppable data meet in the middle of the bit-rate scale providing a good choice both for the customer (cost aspect) and the network provider (cost and utilization aspect).

4. Computations with other statistical features

The advantages of using histogram classification over simple tiling were shown in the previous sections. The classification process can be significantly simplified by the use of other local statistical features such as variance, coefficient of variance, energy, entropy, skewness, kurtosis and image activity [42,43]. These features are described in Table 8.

Additionally, combinations of two or more of these statistical features can also be used in order to ensure the robustness of the neural network classifier in situation where one statistical feature cannot capture the general structure of the distributions of the blocks being classified. The one-input case seems, actually, a little odd and is not usually used in practice although theoretically correct. It is well known that NNs are good at non-linearly separating multidimensional input

Table 8
Statistical features useful for classification

Variance	$\sigma^2 = \frac{1}{N-1}\sum_i (x_i - \bar{x})^2$ *
Coefficient of variance	$n = \frac{\sigma}{\bar{x}}$
Energy	$E = \frac{1}{N}\sum_i x_i^2$
Entropy	$E = -\sum p_x \log_2(p_x)$
Skewness	$S = \frac{1}{N}\sum_i \left(\frac{x_i - \bar{x}}{\sigma}\right)^3$
Kurtosis	$K = \left\{\frac{1}{N}\sum_i \left(\frac{x_i - \bar{x}}{\sigma}\right)^4\right\} - 3$
Image activity	$IAM = \frac{1}{M*N}\left[\sum_{i=1}^{M-1}\sum_{j=1}^{N}\lvert I(i,j) - I(i+1,j)\rvert - \sum_{i=1}^{M}\sum_{j=1}^{N-1}\lvert I(i,j) - I(i,j+1)\rvert\right]$

*where $\bar{x}$ is the mean value.

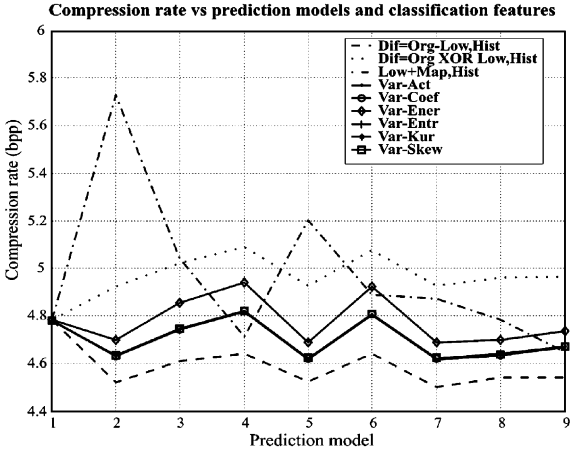

Fig. 12. Compression rate of the system for various prediction models and classification feature-pairs.

signals, and classifiers based on NNs are preferable, in such cases, over VQ due to their reduced complexity. Nevertheless, the classification performs well but the computational cost is relatively high in this case (since the inputs are low-dimensional) and other classification methods should also be considered as good choices.

Fig. 12 shows the average compression rate obtained by the system for various prediction models and various pairs of classification features, over the whole test image set:

The lower dashed line represents the compression rates obtained by using the histograms as classification feature.

- dotted line represents compression rates obtain by encoding the residual image produced by:

$$\text{Residual} = \text{Original XOR LowQuality} \quad (3)$$

  The classification feature is again the block histograms.
- The dotted-dashed line represents compression rates obtained by using histogram classification and context (for the AC) taken both from the classification map and the low-quality image.
- The set of solid marked lines (that coincide in the middle of the graph) represent compression rates obtained when using other combinations of statistical features for classification (such as the combination of variance and entropy).

Results show that a combination of just two statistical entities such as variance + skewness for the classification process appears to give comparable results, while at the same time reduces significantly the complexity of classification.

5. Conclusion and future work

In this work we propose a compression scheme where a histogram classification process decomposes a residual image into segments that can be efficiently and individually encoded by a standard entropy coder, while supporting progressive transmission and lossless reconstruction of the original image. The method works by dividing the residual information to N segments, which are individually encoded. The advantages of the scheme are:

- non-dependence to a specific encoder that produces the low-quality layer,
- segmentation using statistical similarity criteria, which results to the efficient compression of the residual information,
- ability to provide progressive transmission and decoding, and
- better subjective image quality under noisy conditions, due to the progressive transmission capability, which leads to less obvious blocking artifacts,
- ability to impose fees policy on the transmitted data.

The disadvantage of the scheme is that compression is not actually improved in comparison with state-of-the-art lossless compression schemes.

Since the classification process in our scheme is actually a low-complexity VQ algorithm and the problem to be solved is one of clustering image tiles according to local features, other methods can also be employed at this stage. There have already been some preliminary experiments on the usage of LBG, K-means, LVQ and fuzzy c-means [44-51] classification algorithms and evaluating their performance and computational complexity and time in order to choose the best classifier as we are developing and integrating the whole system into a unique coding scheme.

We are currently integrating the system into a new and independent compression scheme based on compression using DWT and application of our block-based segmentation on the high-pass subbands (HL, LH, HH) of the wavelet decomposition. The system is naturally progressive by quality and resolution and there is no need for a decoder in the encoder's side. We are investigating various classification algorithms to choose the one that balances complexity and accuracy. We are also considering the case of migrating to binary arithmetic encoding, following the paradigm of major standards, that will enable us to process data on a bit-level basis. By adopting schemes such as significance coding, the segmented residuals can be further segmented into quality layers within each segment. This way the transmitted packets become even smaller, better manageable and more resilient to random and burst errors. Finally, the system will be generalized for the case of color images by incorporating a reversible color component transform to decorrelate the color channels.

References

[1] B. Carpentieri, M. Weinberger, G. Seroussi, Lossless Compression of Continuous-Tone Images, Hewlett-Packard (2000).

[2] X. Wu, Lossless Compression of Continuous-Tone Images via Context Selection, Quantization and Modeling, IEEE Trans. Image Process. 6 (5) (1997) 656–664.

[3] X. Wu, N. Memon, Lossless Interframe Image Compression via Context Modeling, IEEE Trans. Image Process. 9 (5) (2000) 994–1001.

[4] X. Wu, An Algorithmic Study on Lossless Image Compression, Proceedings of the IEEE Data Compression Conference, Snowbird, Utah, USA, 1996, pp. 150-159.

[5] G. Langdon Jr., Lossless Image Compression—General Overview, cmpe 263 Winter, University of California, Santa Cruz, 2001, Available online at: http://www.cse.ucsc.edu/classes/cmpe263/Winter01/handouts/lossless.pdf.

[6] M.J. Weinberger, G. Seroussi, G. Sapiro, The LOCO-I Lossless Image Compression Algorithm: Principles and Standardization into JPEG-LS, IEEE Trans. Image Process IP- 9 (2000) 1309–1324 Available as Hewlett-Packard Laboratories Technical Report HPL-98-193(R.1).

[7] M. J. Weinberger, G. Seroussi, G. Sapiro, LOCO-I: A low complexity lossless image compression algorithm, ISO/IEC JTC1/SC29/WG1 document N203, July 1995.

[8] X. Wu, N. Memon, Context-based, Adaptive, Lossless Image Coding, IEEE Trans. Commun. 45 (4) (1997) 437–444.

[9] P. G. Howard, J. S. Vitter, Fast and Efficient Lossless Image Compression, Proceedings of the Data Compression Conference, Snowbird, Utah, March 30–April 1, 1993, pp. 351-360.

[10] M. Rabbani, P.W. Jones, Digital Image Compression Techniques, Vol. II, SPIE Optical Engineering Press, 1991.

[11] A. Habibi, Comparison of nth-Order DPCM Encoder With Linear Transformations and Block Quantization Techniques, IEEE Trans. Commun. Tech. COM-19 (1971) 948–956.

[12] K. Jain, Image Data Compression: A Review, Proc. IEEE 69 (1991) 349–389.

[13] N. Netravali, J.O. Limb, Picture coding: a review, Proc. IEEE 69 (1980) 366–406.

[14] J.B. O'Neal, Predictive quantization differential pulse code modulation for the transmission of television signals, Bell Syste. Tech. J. 45 (1966) 689–721.

[15] C. Chamzas, D.L. Duttweiler, Encoding facsimile images for packet-switched networks, IEEE J. Select. Area Commun. 7 (5) (1989) 857–864.

[16] W.B. Pennebaker, J.L. Mitchell, JPEG Still Image Compression Standard, Van Nostrand Reinhold, New York, 1993.

[17] C.S. Burrus, R.A. Gopinath, H. Guo, Introduction to Wavelets and Wavelet Transforms, Prentice-Hall, Englewood Cliffs, NJ, 1998.

[18] I. Daubechies, Orthonormal bases of compactly supported wavelets, Comm. Pure Appl. Math., XLI (1988) 909–966.

[19] I. Daubechies, Ten lectures on Wavelets, SIAM, Philadelphia, PA, 1992.

[20] S. Mallat, A Wavelet Tour of Signal Processing, Academic Press, New York, 1999.

[21] R.M. Rao, A.S. Bopardikar, Wavelet Transforms—Introduction to Theory and Applications, Addison-Wesley, Reading, MA, 1998.

[22] A. Christopoulos, A. Skodras, T. Ebrahimi, The JPEG2000 still image coding system: an overview, IEEE Trans. Consumer Electron. 46 (4) (2000) 1103–1127.

[23] D. Santa-Cruz, T. Ebrahimi, An analytical study of JPEG 2000 functionalities, IEEE ICIP 2000, Vancouver, Canada, 2000, pp. 49-52.

[24] D.S. Taubman, M.W. Marcellin, JPEG 2000: Image Compression Fundamentals, Standards and Practice, Kluwer Academic Publishers, Dordrecht, MA, USA, 2002.

[25] JPEG2000 Special Issue, Signal Processing: Image Communication, Vol. 17, No. 1, Elsevier, Amsterdam, January 2002.

[26] Information Technology—JPEG2000. Image Coding System—Part 1: Core Coding System, International Standard ISO/IEC 15444-1:2000(E), 2000.

[27] A. Moffat, R.M. Neal, I.W. Witten, Arithmetic Coding Revisited, ACM Trans. Inform. Systems 16 (3) (1998).

[28] T. Kohonen, Self-Organizing Maps, Springer, New York, 1997.

[29] T. Kohonen, Self Organization and Associative Memory, 3rd ed, Springer, Berlin-Heidelberg-New York-Tokyo, 1989.

[30] T. Kohonen, The self organizing map, Proc. IEEE 78 (9) (1990) 1464–1480.
[31] T. Kohonen, Self organizing maps, Optimization approaches, Proceedings of the International Conference on Artificial Neural Networks, Espoo, Finland, 1991, pp. 981-990.
[32] S. Haykin, Neural Networks: A Comprehensive Foundation, Macmillan Co., New York, 1994.
[33] R. M. Gray, Fundamentals of Vector Quantization, [online]. Available at http://www-isl.stanford.edu/~gray/compression.html
[34] R.M. Gray, D.L. Neuhoff, Quantization, Proc. IEEE Trans. Inform. Theory 44 (8) (1998).
[35] N. Papamarkos, A. Atsalakis, Gray-Level Reduction Using Local Spatial Features, Comput. Vision Image Understand. 78 (2000) 336–350.
[36] http://links.uwaterloo.ca/bragzone.base.html.
[37] http://www.ijg.org.
[38] http://cygwin.com.
[39] http://www.kakadusoftware.com
[40] G. Pavlidis, A. Tsompanopoulos, N. Papamarkos, C. Chamzas, JPEG2000 over Noisy Communication Channels—The Cost Analysis Aspect, Proceedings of IEEE ICIP 2002, Rochester, New York, USA, September 22–25, 2002.
[41] G. Pavlidis, A. Tsompanopoulos, N. Papamarkos, C. Chamzas, JPEG2000 over Noisy Communication Channels—Thorough Evaluation and Cost Analysis, Signal Processing: Image Communication, Vol. 18, Elsevier, Amsterdam, 2003, pp. 497–514.
[42] S. Saha, R. Vemuri, How do image statistics impact lossy coding performance?, Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'00), Las Vegas, March 27–29, 2000.
[43] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical recipes in C: the art of scientific computing, 2nd ed, Cambridge University Press, Cambridge, 1992, pp. 609-650.
[44] A.K. Jain, M.N. Murty, P.J. Flynn, Data Clustering: A Review, ACM Comput. Surveys 31 (3) (1999) 264–323.
[45] G.W. Milligan, M.C. Cooper, An Examination of Procedures for Determining the Number of Clusters in a Data Set, Psychometrika 50 (1985) 159–179.
[46] D. Fasulo, An Analysis of Recent Work on Clustering Algorithms, Department of Computer Science & Engineering, University of Washington, 1999.
[47] M. de Hoon, S. Imoto, S. Miyano, The C Clustering Library, Institute of Medical Science, Human Genome Center, University of Tokyo (2003).
[48] J.T. Tou, R.C. Gonzalez, Pattern Recognition Principles, Addison-Wesley, Reading, MA, 1974.
[49] M.M. Trivedi, J.C. Bezdek, Low-level segmentation of aerial images with fuzzy clustering, IEEE Trans. Syst. Man. Cybernetics SMC-16 (1986) 589–598.
[50] Y.W. Lim, S.U. Lee, On the color image segmentation algorithm based on the thresholding and the fuzzy c-means techniques, Pattern Recognition 23 (1990) 935–952.
[51] T. Kohonen, J. Kangas, J. Laaksonen, K. Torkkola, LVQ_PAK: A Program Package for the Correct Application of Learning Vector Quantization Algorithms, Proceedings of the IEEE International Joint Conference on Neural Networks, Baltimore Vol. I (1992) 725–730.