ELSEVIER

Signal Processing: *Image Communication* 17 (2002) 113–130

SIGNAL PROCESSING:
IMAGE
COMMUNICATION

www.elsevier.com/locate/image

JPEG 2000 performance evaluation and assessment

Diego Santa-Cruz*, Raphaël Grosbois, Touradj Ebrahimi

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

Abstract

JPEG 2000, the new ISO/ITU-T standard for still image coding, has recently reached the International Standard (IS) status. Other new standards have been recently introduced, namely JPEG-LS and MPEG-4 VTC. This paper provides a comparison of JPEG 2000 with JPEG-LS and MPEG-4 VTC, in addition to older but widely used solutions, such as JPEG and PNG, and well established algorithms, such as SPIHT. Lossless compression efficiency, fixed and progressive lossy rate-distortion performance, as well as complexity and robustness to transmission errors, are evaluated. Region of Interest coding is also discussed and its behavior evaluated. Finally, the set of provided functionalities of each standard is also evaluated. In addition, the principles behind each algorithm are briefly described. The results show that the choice of the "best" standard depends strongly on the application at hand, but that JPEG 2000 supports the widest set of features among the evaluated standards, while providing superior rate-distortion performance in most cases. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: JPEG; JPEG 2000; Image compression; MPEG; JPEG-LS; Wavelet; Lossless; MPEG-4

1. Introduction

The standardization effort of the next ISO/ITU-T standard for compression of still images, JPEG 2000 [9], has recently reached the International Standard (IS) status, 4 years after the call for proposals [10]. Great efforts have been made by all the participants to deliver a new standard for today's and tomorrow's applications, by providing features inexistent in previous standards, but also by providing higher efficiency for features that exist in others. Now that the standard has been finalized and accepted some trivial questions would be: how well does it perform, what are the features offered by JPEG 2000 and how well are they fulfilled, when compared to other standards offering the same or similar features. This paper aims at providing an answer to this simple but somewhat complex question. Section 2 provides a brief overview of the techniques compared, Section 3 evaluates different aspects of compression efficiency, while Section 4 looks at the algorithms' complexities. Error resilience performance is studied in Section 5 and functionality in Section 6. Finally conclusions are drawn in Section 7.

2. Overview of evaluated algorithms

For the purpose of this study we compare the coding algorithm in the JPEG 2000 standard

---

*Corresponding author.
E-mail address: diego.santacruz@epfl.ch (D. Santa-Cruz).

PII: S0923-5965(01)00025-X

to the following three standards: JPEG [17], MPEG-4 Visual Texture Coding (VTC) [7] and JPEG-LS [6]. In addition, we also include SPIHT [20] and PNG [25]. The reasons behind this choice are as follows. JPEG is one of the most popular coding techniques in imaging applications ranging from Internet to digital photography. Both MPEG-4 VTC and JPEG-LS are very recent standards that start appearing in various applications. It is only logical to compare the set of features offered by the JPEG 2000 standard not only to those offered in a popular but older one (JPEG), but also to those offered in most recent ones using newer state-of-the-art technologies. SPIHT is a well-known representative of state-of-the-art wavelet codecs and serves as a reference for comparison. Although PNG is not formally a standard and is not based on state-of-the-art techniques, it is becoming increasingly popular for Internet-based applications. PNG is also undergoing standardization by ISO/IEC JTC1/SC24 and will eventually become ISO/IEC international standard 15948. Similar, less complete, evaluations have been previously presented in [22,21,13].

2.1. JPEG

This is the very well known ISO/ITU-T standard created in the late 1980s. There are several modes defined for JPEG [17,26], including baseline, lossless, progressive and hierarchical. The baseline mode is the most popular one and supports lossy coding only. The lossless mode is not popular but provides for lossless coding, although it does not support lossy.

In the baseline mode, the image is divided in 8×8 blocks and each of these is transformed with the Discrete Cosine Transform (DCT). The transformed blocks' coefficients are quantized with a uniform scalar quantizer, zig-zag scanned and entropy coded with Huffman coding. The quantization step size for each of the 64 DCT coefficients is specified in a quantization table, which remains the same for all blocks. The DC coefficients of all blocks are coded separately, using a predictive scheme. Hereafter we refer to this mode simply as JPEG.

The lossless mode is based on a completely different algorithm, which uses a predictive scheme. The prediction is based on the nearest three causal neighbors and seven different predictors are defined (the same one is used for all samples). The prediction error is entropy coded with Huffman coding. Hereafter we refer to this mode as L-JPEG.

The progressive and hierarchical modes of JPEG are both lossy and differ only in the way the DCT coefficients are coded or computed, respectively, when compared to the baseline mode. They allow a reconstruction of a lower quality or lower resolution version of the image, respectively, by partial decoding of the compressed bitstream. Progressive mode encodes the quantized coefficients by a mixture of spectral selection and successive approximation, while hierarchical mode uses a pyramidal approach to computing the DCT coefficients in a multi-resolution way.

2.2. JPEG-LS

JPEG-LS [6] is the latest ISO/ITU-T standard for lossless coding of still images, which in addition provides support for “near-lossless” compression. The main design goal of JPEG-LS has been to deliver a low-complexity solution for lossless image coding with a compression efficiency that is very close to the best reported results, which have been obtained with much more complex schemes, such as CALIC [28]. As a design trade-off, much of the extra functionality that is found in other standards, such as support for scalability and error resilience, has been left out.

JPEG-LS, which is based on the LOCO-I [27] algorithm, is composed of two parts: part I specifies the baseline system, which is suitable for most applications, while part II defines extensions which improve the coding efficiency for particular types of imagery at the expense of increased complexity. Part I is based on adaptive prediction, context modeling and Golomb coding. In addition it features a flat-region detector to encode these in run-lengths. Part I also supports “near-lossless” compression by allowing a fixed maximum sample error. Part II introduces an arithmetic coder, which is intended to deal with the limitations that part I presents when dealing with very compressible images (i.e. computer graphics, images with sparse histograms).

2.3. MPEG-4 VTC

MPEG-4 Visual Texture Coding (VTC) [23,12] is the algorithm used in MPEG-4 [7,18,5] to compress visual textures and still images, which are then used in photo realistic 3D models, animated meshes, etc., or as simple still images. It is based on the discrete wavelet transform (DWT), scalar quantization, zero-tree coding and arithmetic coding. The DWT is dyadic and uses a Daubechies (9,3) taps biorthogonal filter [2].

The quantization is scalar and can be of three types: single (SQ), multiple (MQ) and bi-level (BQ). With SQ each wavelet coefficient is quantized once, the produced bitstream not being SNR scalable. With MQ a coarse quantizer is used and this information coded. A finer quantizer is then applied to the resulting quantization error and the new information coded. This process can be repeated several times, resulting in limited SNR scalability. BQ is essentially like SQ, but the information is sent by bit-planes, providing general SNR scalability.

Two scanning modes for the generated wavelet coefficients are available: tree-depth (TD), the standard zero-tree scanning, and band-by-band (BB). The former produces a SNR scalable bitstream only, while the latter produces a resolution progressive one where the data is SNR scalable within each resolution level if MQ or BQ is used.

A unique feature of MPEG-4 VTC is the capability to code arbitrarily shaped objects. This is accomplished by the means of a shape adaptive DWT and MPEG-4's shape coding. Several objects can be encoded separately, possibly at different qualities, and then composited at the decoder to obtain the final decoded image. On the other hand, MPEG-4 VTC does not support lossless coding.

2.4. PNG

Portable Network Graphics (PNG) [25] is a W3C recommendation for coding of still images which has been elaborated as a patent-free replacement for GIF, while incorporating more features than this last one. It is based on a predictive scheme and entropy coding. The prediction is done on the three nearest causal neighbors and there are five predictors that can be selected on a line-by-line basis. The entropy coding uses the Deflate algorithm of the popular Zip file compression utility, which is based on LZ77 coupled with Huffman coding. PNG is capable of lossless compression only and supports gray scale, paletted color and true color, an optional alpha plane, interlacing and other features.

2.5. SPIHT

Although not a standard or a widely used scheme in user applications, set partioning in hierarchical trees (SPIHT), the algorithm introduced by Said and Pearlman [20], has become a widely known one in the image coding community and makes a good candidate as a basis to compare wavelet-based coders from the compression efficiency point of view. Hence, its inclusion in this comparative study.

SPIHT is based on the DWT and exploits its self-similarity across scales by using set partitioning. The wavelet coefficients are ordered into sets using the parent–child relationship and their significance at successively finer quantization levels. The binary decisions can be further compressed by the use of an optional arithmetic coder. The produced bitstream is SNR scalable only. This algorithm is capable of lossless as well as lossy compression. For the former the reversible S + P filter [19] is used while for the latter the non-reversible Daubechies (9,7) taps biorthogonal one [2] is used, which provides higher compression and is also used in JPEG 2000.

2.6. JPEG 2000

Although the algorithm behind JPEG 2000, as well as its features and functionality set, are duly described in the other articles of this special issue we provide here a brief overview for the sake of completeness. We restrict ourselves to part 1 of the standard [9] only, which defines the core system.

JPEG 2000, which is largely derived from EBCOT [24], is based on the DWT, scalar quantization, context modeling, arithmetic coding and post-compression rate allocation. The DWT is dyadic and can be performed with either the reversible Le Gall (5,3) taps filter [11], which provides for lossless coding, or the non-reversible Daubechies (9,7) taps biorthogonal one [2], which provides for higher compression but does not do lossless. The quantizer follows an embedded dead-zone scalar approach and is independent for each subband. Each subband is divided into rectangular blocks (called codeblocks in JPEG 2000), typically 64 coefficients wide and 64 tall, and entropy coded using context modeling and bit-plane arithmetic coding. The coded data is organized in so called layers, which are quality levels, using the post-compression rate allocation and output to the codestream in packets. The generated codestream is parseable and can be resolution, layer (i.e. SNR), position or component progressive, or any combination thereof.

JPEG 2000 also supports a number of functionalities, many of which are inherent from the algorithm itself. Examples of this is random access, which is possible because of the independent coding of the code-blocks and the packeti-zed structure of the codestream. Another such functionality is the possibility to encode images with arbitrarily shaped Regions of Interest (ROI) [4]. The fact that the subbands are encoded bit-plane by bit-plane makes it possible to select regions of the image that will precede the rest in the codestream by scaling the subband samples so that the bit-planes encoded first only contain ROI information while following ones only contain background information. The only thing the decoder needs to receive is the factor by which the samples were scaled. The decoder can then invert the scaling based only on the amplitude of the samples. Other supported functionalities are error-resilience, multicomponent images, palletized color, compressed domain lossless flipping and simple rotation, to mention a few.

3. Compression performance

When evaluating image-coding algorithms, there are several factors that determine the choice of a particular algorithm for an application. An important one in most cases is compression efficiency, which we evaluate in this section. However, there are other factors, such as the set of provided functionality and the complexity, that can be even more determining than pure compression efficiency. Those will be evaluated in subsequent sections.

Compression efficiency is evaluated for lossless and lossy compression. In the first case, it is simply measured by the achieved compression ratio for each one of the images. For lossy coding we measure the peak signal-to-noise ratio (PSNR) of the decoded image with respect to the original, defined as $-20\log_{10}(\sqrt{\mathrm{MSE}}/(2^b-1))$, where b is the bit depth and MSE is the mean square error between the two images. In the case where averages across several images are presented we use the average MSE, instead of the MSE of a single image, to obtain a PSNR measure.

The evaluation has been done on seven images from the official JPEG 2000 test set, covering various types of imagery. The images “bike” (2048 × 2560) and “cafe” (2048 × 2560) are natural, “cmpnd1” (512 × 768) and “chart” (1688 × 2347) are compound documents consisting of text, photographs and computer graphics, “aerial2” (2048 × 2048) is an aerial photography, “target” (512 × 512) is a computer-generated image and “us” (512 × 448) an ultra scan. All these images have a depth of 8 bits per pixel.

The results have been generated on a PC with a 550 MHz Pentium$^{\mathrm{TM}}$ III processor, 512 kB of half-speed L2 cache and 512 MB of RAM (SDRAM) under Linux 2.2.12. The software implementations used for coding the images are: the JPEG 2000 Verification Model (VM) 6.1 (ISO/IEC JTC1/SC29/WG1 N 1580), the MPEG-4 MoMuSys VM of Aug. 1999 (ISO/IEC JTC1/SC29/WG11 N 2805), the Independent JPEG Group (IJG) JPEG implementation (http://www.ijg.org/), version 6b, the SPMG JPEG-LS implementation of the University of British Columbia (http://spmg.ece.ubc.ca/), version 2.2, the Lossless JPEG codec of Cornell University (ftp://ftp.cs.cornell.edu/pub/multimed), version 1.0, the libpng implementation of PNG (ftp://ftp.uu.net/graphics/png), version 1.0.3, and the SPIHT codecs, version 8.01, (http://www.cipr.rpi.edu/research/SPIHT/). In the case of SPIHT only the codecs with arithmetic coding have been used.

3.1. Lossless compression

Fig. 1 shows the lossless compression efficiency of JPEG 2000, JPEG-LS, lossless JPEG (L-JPEG), PNG and SPIHT. For JPEG 2000 the reversible (5,3) DWT filter has been used. In the case of L-JPEG optimized Huffman tables and the predictor yielding the best compression performance have been used for each image. For PNG the maximum compression setting has been used, while for JPEG-LS the default options were chosen. For SPIHT the S + P transform was used. MPEG-4 VTC is not considered, as it does not provide a lossless functionality.

It can be seen that in the majority of cases the best performance is obtained by JPEG-LS. JPEG 2000 provides, in most cases, competitive compression ratios with the added benefit of scalability. SPIHT and PNG performance is very close to the one of JPEG 2000 on most images. As for lossless JPEG, it does not perform as well as the other,

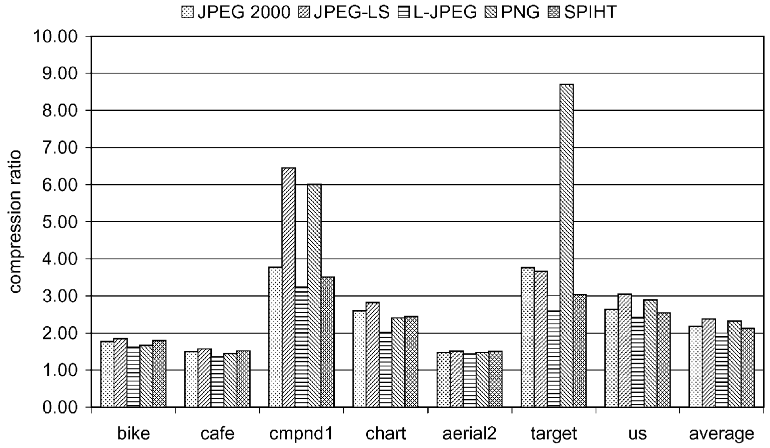

Fig. 1. Lossless compression ratios obtained for each image, as well as the average.

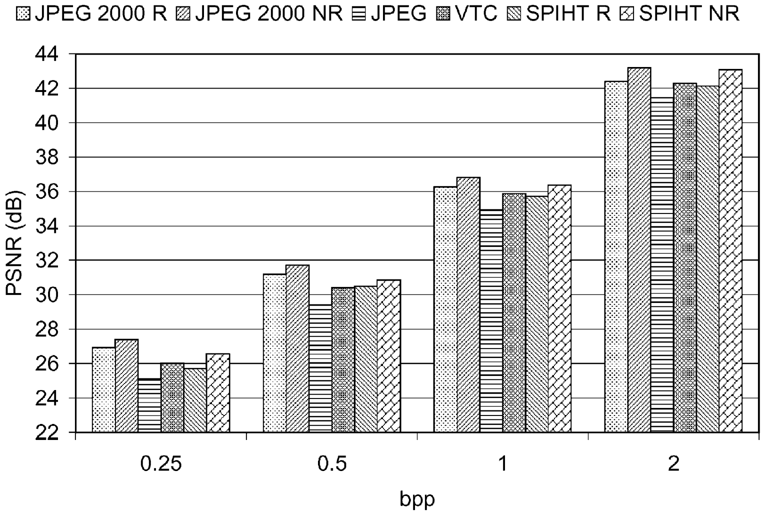

Fig. 2. Non-progressive compression efficiency, for various bitrates, as PSNR of the average MSE across all tested images.

more recent, standards. An important exception to the general trend is the "target" image and, to a lesser extent, "cmpnd1". The former is a computer generated image composed of patches of constant gray level as well as gradients, while the latter is mostly black text on a white background composed with a natural image. For computer-generated images such as "target" PNG provides the best results, probably because of LZ77 being able to exploit the regular structure. In average PNG performs the best, although this is solely due to the very large compression ratio it achieves on "target". However, JPEG-LS provides the best compression ratio for most images.

This shows that as far as lossless compression is concerned, JPEG 2000 seems to perform reasonably well in terms of its ability to efficiently deal with various types of images. Furthermore, when compared to the other wavelet-based scheme (SPIHT) it performs sensibly better on non-natural images and thus is more flexible. However, in specific types of images such as "cmpnd1" and "target" JPEG 2000 is outperformed by far in JPEG-LS. This result is even more striking noting that JPEG-LS is a significantly less complex algorithm.

3.2. *Non-progressive lossy compression*

Fig. 2 shows the compression efficiency of the evaluated algorithms capable of lossy compression, when non-progressive bitstreams are produced.[1] For each bitrate, a single bitstream is generated and decoded in its entirety. For JPEG 2000, results for reversible (5,3) and non-reversible (9,7) DWT filters are shown, referred to as JPEG 2000 R and JPEG 2000 NR, respectively. Similarly, for SPIHT results for the reversible S + P transform and the non-reversible (9,7) DWT are shown, referred to as SPIHT R and SPIHT NR. In the case of JPEG, the baseline mode has been used with flat quantization tables and optimized Huffman tables. The results shown, which are averages across all tested images, are representative of individual ones too.

From the results it is clear that JPEG 2000 with the non-reversible filter outperforms all other algorithms at all bitrates. The reversible filter pays a small penalty for its capability of performing a lossless decoding, but it still outperforms all other algorithms, with the solely exception of SPIHT with the non-reversible filter at 2 bpp where it is slightly worse. Having a lossless image after decoding can be of significant value to many applications (archiving, medical, etc.). JPEG provides, as expected for older technology, inferior results, showing a considerable quality difference at any given bitrate. It is also worth noting that the advantage of JPEG 2000 over the others gets larger with increasing compression ratios.

3.3. *SNR progressive lossy compression*

Fig. 3 shows the compression efficiency in the case of SNR progressive bitstreams across various bitrates. In this case one bitstream has been generated at 2 bpp for each algorithm and then decoded at 0.25, 0.5, 1 and 2 bpp. The acronyms JPEG 2000 R, JPEG 2000 NR, SPIHT R, SPIHT NR refer to the same techniques as in the previous section. For MPEG-4 VTC, the results have been generated using multiple quantization (MQ) and targeting the tested bitrates. In the case of JPEG the progressive mode has been used where the coefficients are encoded by successive refinement, and is referred to as P-JPEG. As before the results shown, which are averages across all tested images, are representative of individual ones.

As in the non-progressive case, the results show clearly that JPEG 2000 with the non-reversible filter outperforms all other algorithms. In fact, comparing against the results of Fig. 2 one can note that all the algorithms, besides JPEG, pay little or no penalty when producing progressive bitstreams.[2] On the contrary, JPEG suffers a considerable degradation when progressive bitstreams are generated and thus is outperformed by

---

[1] By the nature of the algorithm SPIHT always generates SNR progressive bitstreams, although that fact is not exploited in this test.

[2] MPEG-4 VTC can in some cases appear to perform better when progressive bitstreams are produced. This stems from the fact that with this algorithm it is not possible to target precise bitrates and these small variations can account for the PSNR difference in the graphs.

far by the other algorithms. Also, as in the non-progressive case, the advantage of JPEG 2000 over the others gets larger as the compression ratio increases.

3.4. ROI coding

Region of Interest (ROI) coding is one of the novel functionalities of JPEG 2000. There are several ROI modes in JPEG 2000, although part-1 only supports the so called *maxshift* method, which was briefly explained in Section 2.6. While the maxshift method supports arbitrarily shaped regions it requires that all ROI information be sent prior to any background one. The more general scaling based method of part-2 lifts this restriction but allows only predefined ROI shapes. In the following we restrict our evaluation to the maxshift method.

Fig. 4 shows the decoded PSNR of ROI and background (BG) independently for the "bike" image. The ROI's upper-left corner is located at

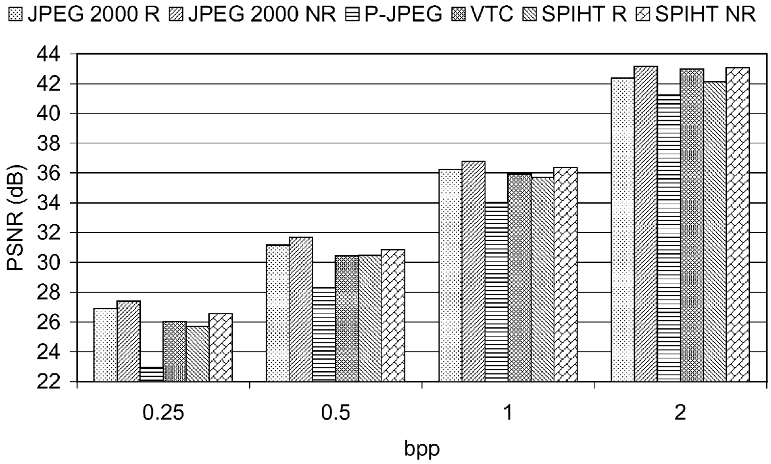

Fig. 3. SNR progressive compression efficiency, for various bitrates, as PSNR of the average MSE across all tested images.

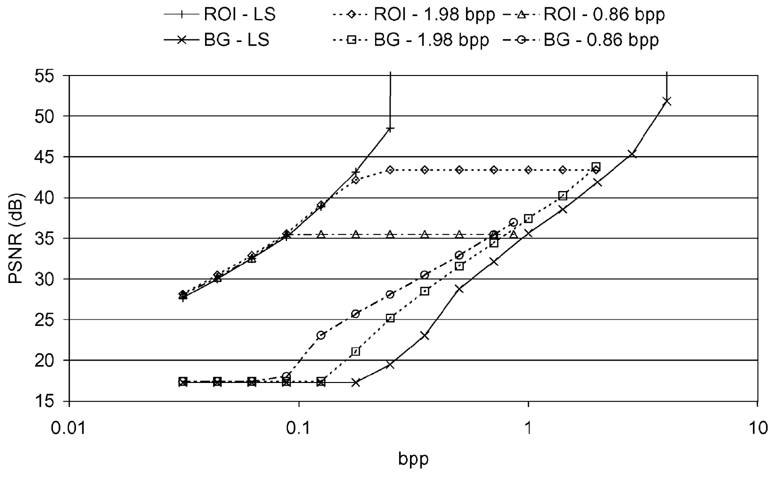

Fig. 4. ROI behaviour at various overall coded bitrates, for the "bike" image. The overall bitrate for lossless (LS) coding is 4.58 bpp.

(600, 2130) and its width and height is 750 and 350 pixels, respectively, thus covering 5% of the image. Three SNR progressive bitstreams were generated, one lossless (LS) at 4.58 bpp, and two lossy, with the (9,7) filter, at 1.98 and 0.86 bpp. The bitstreams were then decoded at various bitrates ranging from 0.03125 to the maximum rate. In order to provide a low-resolution version of background context together with the ROI, the subbands of the first (i.e. lowest) and second resolution levels were entirely associated with the ROI. In this case the JPEG 2000 implementation used was JJ2000 (http://jj2000.epfl.ch), version 4.0, contained in the JPEG 2000 part-5 FCD [8], since it provides better ROI support than the VM.

The results show that ROI and BG quality increase independently as more data is decoded. The ROI quality improves from the beginning, while the BG one starts improving only after the ROI has been almost entirely transmitted. The curves also show the tradeoff that exists between the maximum ROI quality and the bitrate at which the BG starts improving. Fig. 5 shows the visual results for the bitstream encoded at 1.98 bpp. It shows a portion of the decoded image at various stages: very low rate (0.03125 bpp), where the ROI is already understandable; when the ROI is almost entirely decoded (0.177 bpp); when the BG is almost entirely decoded (0.5 bpp) and when the bitstream is completely decoded (1.98 bpp). This clearly shows the advantage of ROI coding: in a progressive transmission, the ROI is received at a good quality much earlier than if no ROI coding was used, while the BG remains available at a good quality as well. Note also the graceful degradation from the ROI to the BG.

3.5. Visual quality

It is a very well known fact that MSE and PSNR do not always relate well to visual quality, and medium and small differences in PSNR do not necessarily mean that the visual quality difference is in the same direction, in particular when radically different coding algorithms, with different artifact types, are compared. For this reason, in this section we report results from visual comparisons.

Within WG1 a study comparing the visual quality of JPEG and JPEG 2000 has been performed [3]. In this study, six color images

Fig. 5. Magnified portions of the "bike" image, encoded at 1.98 bpp, with a ROI, and decoded at 0.03125, 0.177, 0.5 and 1.98 bpp, in clockwise order. The ROI occupies the lower-right half of the image, approximately.

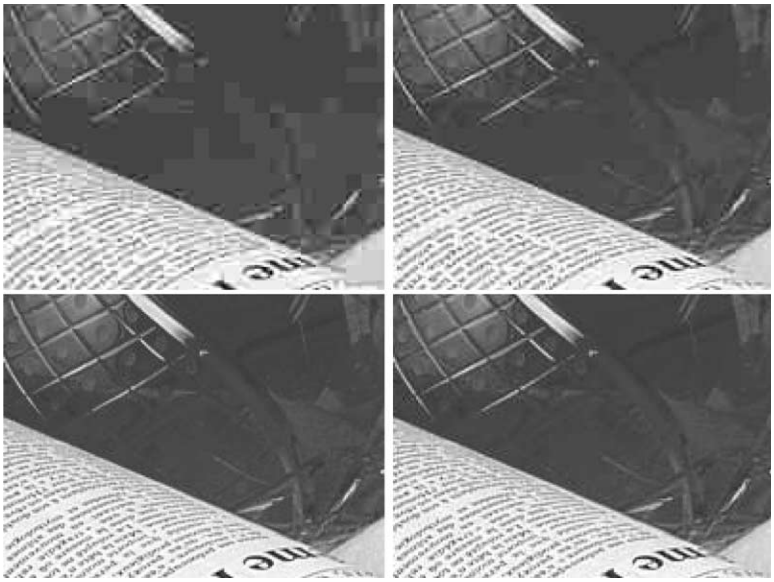

Fig. 6. Magnified portion of the "bike" image when compressed with JPEG at various bitrates. From the top-left corner, in clokwise order the bitrates are: 0.25, 0.5, 1.0 and 1.5 bpp.

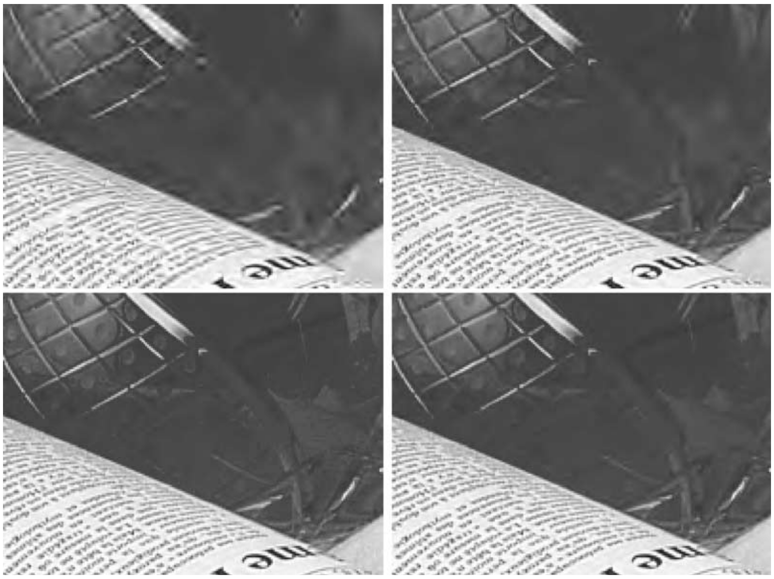

Fig. 7. Magnified portion of the "bike" image when compressed with JPEG 2000 at various bitrates. From the top-left corner, in clokwise order the bitrates are: 0.25, 0.5, 1.0 and 1.5 bpp.

compressed at various bitrates with JPEG and JPEG 2000 have been ranked by six observers using high-quality hardcopies. For both algorithms visual weighting has been applied to improve the visual quality. The results show that, in average, JPEG needs 112%, 55%, 22%

and 13% more bitrate to match the visual quality of JPEG 2000 at 0.25, 0.5, 0.75 and 1 bpp, respectively. This confirms the conclusions drawn from the results of Sections 3.2 and 3.3: the advantage in compression efficiency of JPEG 2000 over JPEG gets larger as the compression ratio increases. In addition, most images coded with JPEG 2000 were found to be visually lossless at 1 bpp. Regarding artifacts, JPEG 2000 does not suffer from color distortions and there is no blockiness. However, image sharpness tends to increase faster with JPEG than with JPEG 2000, as bitrate increases.

As an example, Figs. 6 and 7 show a portion of the "bike" image coded with JPEG and JPEG 2000 at various bitrates. One can see that JPEG 2000 quality is clearly superior at low bitrates (e.g. 0.25 and 5 bpp), although both algorithms present visible artifacts. At 1.0 bpp, JPEG shows small artifacts around the edge of the newspaper, while JPEG 2000 shows no visible ones.

4. Complexity

Besides compression efficiency, another important aspect of an image compression system is the complexity of the algorithm execution. However, depending on the application and the working environment, the complexity depends on different factors. They can be memory bandwidth, total working memory, number of arithmetic operations, number of hardware gates, etc. Furthermore, these numbers are very dependent on the optimization, targeted applications and other factors of the different implementations. A thorough complexity analysis of the different algorithms is beyond the scope of this article. However, run times of the different programs implementing the algorithms on a PC like platform provide an appreciation of the involved complexity for many of the potential applications. More detailed analysis of the various parts of the JPEG 2000 algorithm's complexity can be found in [15,16].

Figs. 8 and 9 show the execution times for lossless compression and decompression, as microseconds per pixel, for the results reported in Section 3.1. One can observe that JPEG-LS, in addition to providing the best compression ratios, also provides the fastest compression and is presumably the least complex. JPEG 2000 is considerably more complex, although its compression efficiency is not as high as JPEG-LS. The decompression times are similar to the compression ones, with the notable exception of

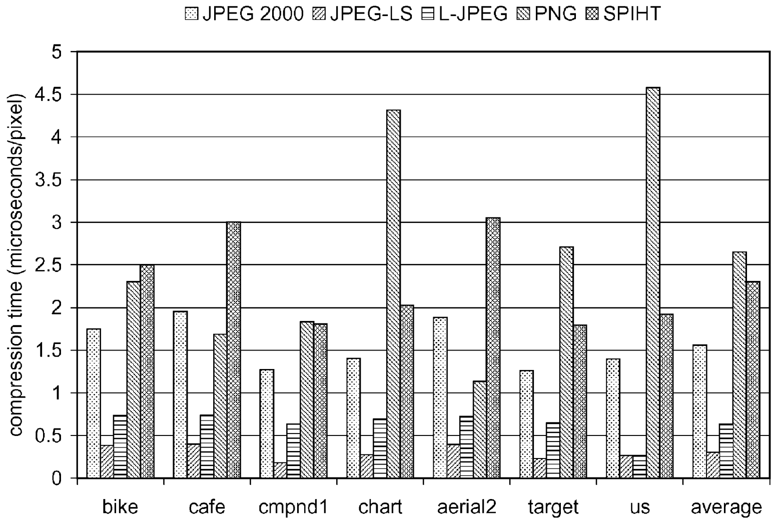

Fig. 8. Lossless compression execution times, in microseconds per pixel, for various images and their average.

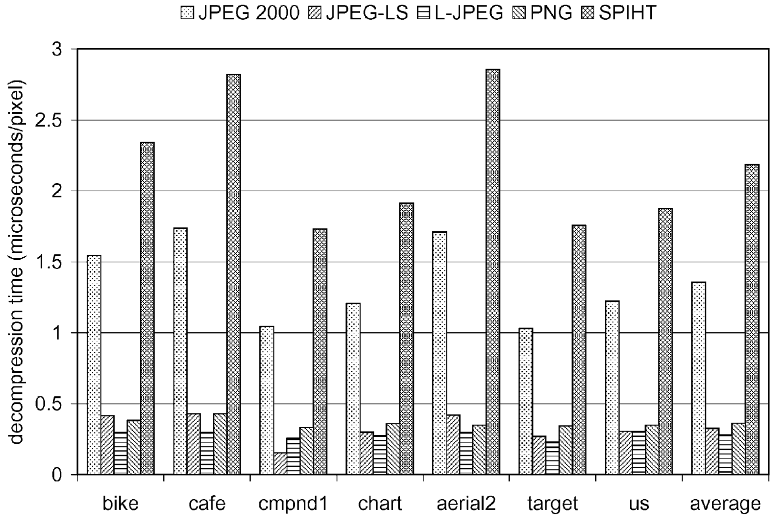

Fig. 9. Lossless decompression execution times, in microseconds per pixel, for various images and their average.

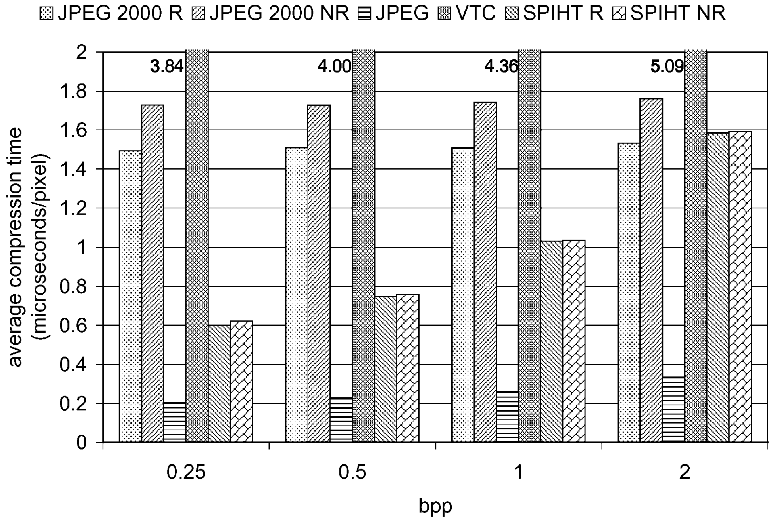

Fig. 10. Non-progressive compression execution times, in microseconds per pixel, for various bitrates. The values presented are averages across all images. MPEG-4 VTC is off-scale.

PNG and, to a lesser extent, L-JPEG. In fact, PNG being a dictionary-based algorithm, is highly asymmetrical and decompression times are similar to the ones of JPEG-LS, while compression ones are very much longer. L-JPEG also shows asymmetry, although this is due to the fact that optimized Huffman tables are used, requiring a double scan. If the default Huffman tables are used the compression time can be reduced, at the expense of degraded compression ratios. SPIHT exhibits long compression as well as decompression times. It should also be noted that, besides PNG at compression, the results are consistent among the different images, showing that the relative complexity of the algorithms is essentially independent of the image type.

Figs. 10 and 11 show the average execution times, as microseconds per pixel, for the results of

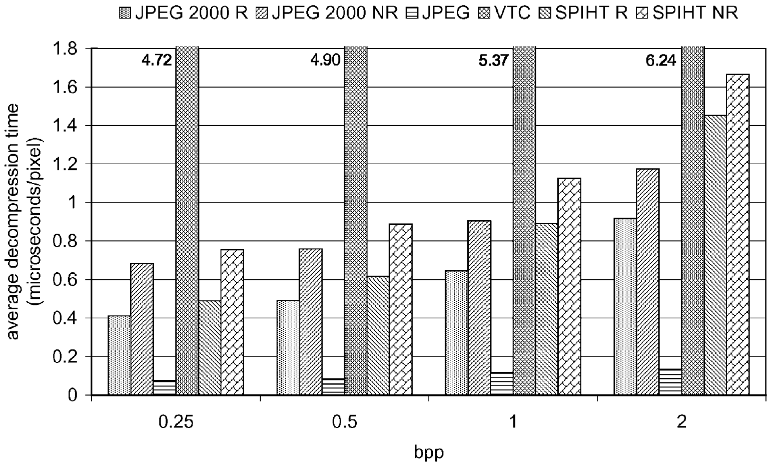

Fig. 11. Non-progressive decompression execution times, in microseconds per pixel, for various bitrates. The values presented are averages across all images. MPEG-4 VTC is off-scale.

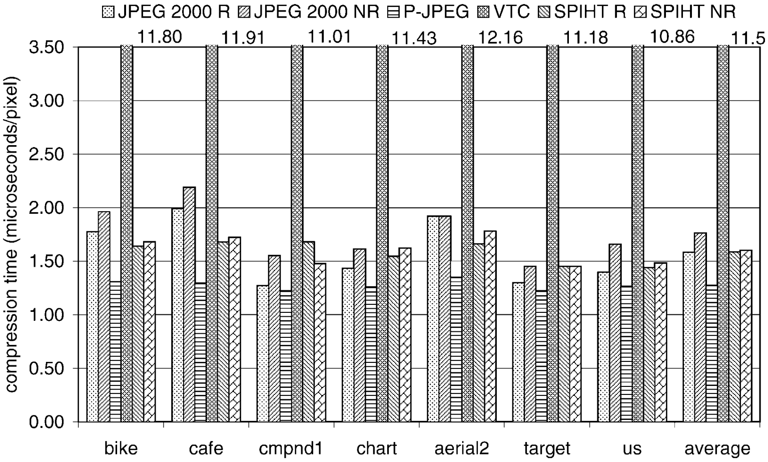

Fig. 12. SNR progressive compression times, in microseconds per pixel, at 2 bpp. MPEG-4 VTC is off-scale.

Section 3.2. As it can be seen, JPEG 2000, MPEG-4 VTC and SPIHT are all significantly slower than JPEG at both compression and decompression. In particular, MPEG-4 VTC is extremely slow without providing any better compression than JPEG 2000 or SPIHT. This could be due to badly written software and these numbers should be taken as a rough indicative measure only. One surprising result is that JPEG 2000 compression time is not dependent on bitrate. The reason for this is that the JPEG 2000 VM encoder compresses the image to a fairly high bitrate, using a small quantization step size, and discard the excess information at the post-compression stage only. The compression time for low bitrates could be improved by choosing larger quantization step sizes without affecting the quality. One can also observe that, in JPEG 2000, the reversible (5,3) filter leads to significantly faster execution than the non-reversible (9,7) one. This can be explained by the fact

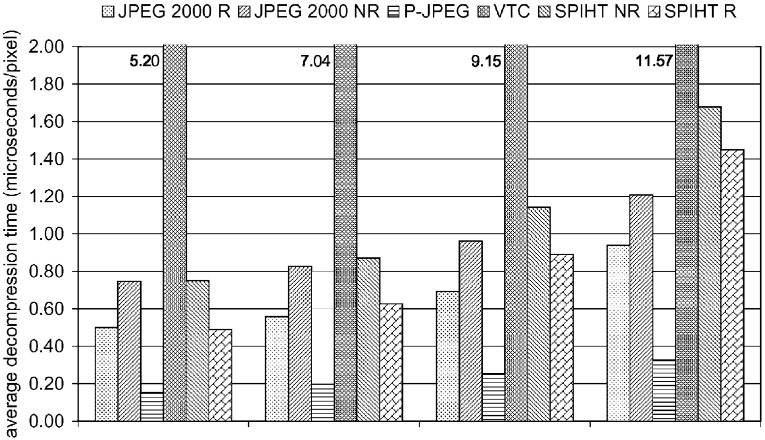

Fig. 13. SNR progressive decompression times, in microseconds per pixel, for various bitrates. MPEG-4 VTC is off-scale.

that the (5,3) is a shorter filter, implemented in integer arithmetic, involving additions, subtractions and shifts only. On the contrary, the (9,7) is larger, implemented in floating-point, and with non-trivial coefficients.

Finally, Figs. 12 and 13 show the compression and decompression times for the SNR progressive results of Section 3.3. A behavior similar to the non-progressive case can be observed for most algorithms. However, progressive JPEG appears to be more complex than baseline JPEG, in particular for higher bitrates (0.5–2 bpp). This is due to the fact that progressive JPEG requires several scans. On the other hand, JPEG 2000 and SPIHT do not exhibit any significant execution overhead when dealing with progressive bitstreams. MPEG-4 VTC shows a significant increase in execution time between non-progressive and progressive, probably due to the change between single quantization (SQ) and multiple quantization (MQ). From Fig. 12, one can also observe that the results are consistent between the different images.

5. Error resilience

In the recent years there has been a sharp increase on the amount of data that is transmitted across wireless networks. Such networks are, in general, error-prone and require, from the image coding system, techniques to protect the data, detect errors and recover from them. In fact, even if at the network level the protocols provide for error protection and concealment, either there is a non-negligible residual error or the overhead of such protections is too large. JPEG and JPEG-LS provide basic error resilience mechanisms, while JPEG 2000 and MPEG-4 VTC contain more advanced ones. PNG has not been designed for wireless applications and only provides support for error detection, but not concealment. SPIHT is not considered here, since it is a pure compression algorithm which does not provide any error resilience tools, although extensions of it exist that include them [1].

In the following we have simulated a symmetric binary transmission channel with random errors and evaluated the average reconstructed image quality after decompression for JPEG baseline and JPEG 2000. MPEG-4 VTC and JPEG-LS could not be evaluated since the software did not offer proper error resilience support. For a review of the error resilience techniques of JPEG 2000 and MPEG-4 VTC, as well as an evaluation of their performance, the reader is directed to [14]. Fig. 14 shows the average results, over 200 runs, for the cafe image for bit error rates of 10^{-6}, 10^{-5} and 10^{-4}, as well as for no errors, for various bitrates. In the case of JPEG the results have been

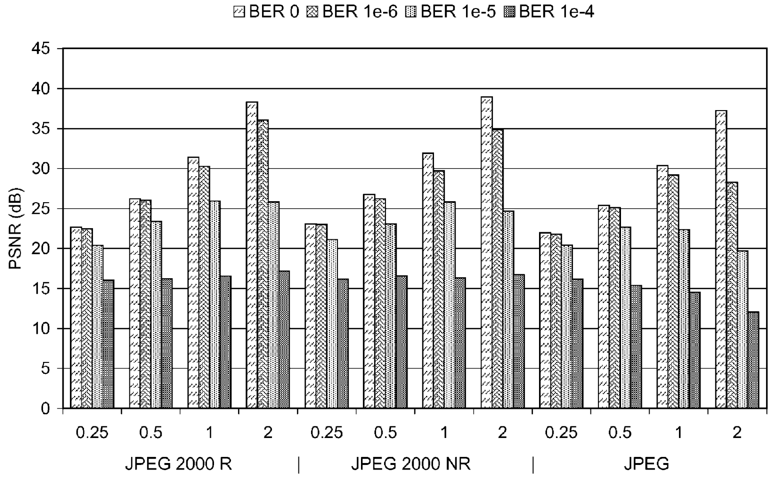

Fig. 14. Average decoded quality of the "cafe" image for transmission bit error rates of 10^{-6}, 10^{-5} and 10^{-4} as well as no errors, when the bitstreams are protected with error resilient techniques.

Fig. 15. Magnified portion of the "cafe" image coded with JPEG at 0.5 bpp, when transmitted over a channel with bit error rates of 10^{-4} (left) and 10^{-5} (right). The images shown are the ones with median MSE, of 200 runs.

obtained by using the maximum amount of restart markers, which amounts to an overhead of less than 1%. In the case of JPEG 2000 the sensitive packet head information has been moved to the bitstream header (using a PPM marker) and the entropy coded data has been protected by the regular termination of the arithmetic coder combined with the error resilient termination and segment symbols. The overhead of these protections amount also to less than 1%. In both cases the bitstream header is transmitted without errors.

As it can be seen, the reconstructed image quality under transmission errors is higher for JPEG 2000 than for JPEG, across all encoding bitrates and error rates. Although both suffer from severe degradation at moderately high error rates (i.e. 10^{-4} and 10^{-5}), at lower ones (i.e. 10^{-6}) JPEG 2000 proves to be fairly robust. One can also observe that at high error rates (10^{-4}) the reconstructed image quality in JPEG 2000 increases very little with increasing bitrate. The reason for this is that in JPEG 2000 each subband block is coded by bit-planes. When the error rate is high enough, almost all code-blocks are affected by an error in its most significant bit-planes, and the lower ones cannot be decoded. Thus, increasing the bitrate does not help to increase the

Fig. 16. Magnified portion of the "cafe" image coded with JPEG 2000 at 0.5 bpp, when transmitted over a channel with bit error rates of 10^{-4} (left) and 10^{-5} (right). The images shown are the ones with median MSE, of 200 runs.

decoded image quality since the extra information cannot be decoded. In the case of JPEG the problem is even worse: the higher the encoding bitrates the lower the decoded quality. This can be explained by the fact that in JPEG the error protection density is a constant per block, but as the bitrate increases the length of each coded block increases and therefore the error protection density per bit decreases. It is also worth noting that in JPEG 2000 the reversible (5,3) filter is less sensitive to errors than the non-reversible (9,7) one. A possible explanation might be that since the filters are shorter the area affected by an error is smaller.

Figs. 15 and 16 show that the decoded visual quality under various bit error rates is higher for JPEG 2000 than for JPEG. In fact, the artifacts created by transmission errors in JPEG 2000 are of the same nature as those introduced by quantization. In JPEG, when a transmission error occurs it is often the entire 8 × 8 block which is missing and/or misplaced and the rest of the strip will also be affected. In some cases, even the bottom of the image will be missing as well.

6. Supported functionality

Besides compression efficiency and complexity, many applications benefit or even require other features which may determine the choice of a particular algorithm for an application. These extra features are often referred to as the set of supported functionality. Examples of such features are the ability to distribute quality in a non-uniform fashion across the image (e.g. ROI), resilience to residual transmission errors, random access to different regions of the coded image, scalability of the generated bitstream, etc.

Table 1 summarizes the results of the comparison of different algorithms from a functionality point of view. SPIHT has been omitted since it is not a complete image coding system. The table clearly shows that from this perspective, JPEG 2000 is the standard offering the richest set of features in an efficient manner and within an integrated algorithmic approach. Although some of the rows in this table are self-explanatory, others deserve some comments.

Both MPEG-4 VTC and JPEG 2000 are able to produce progressive bitstreams without any noticeable overhead, unlike JPEG. However, JPEG 2000 provides more progressive types and orderings than MPEG-4 VTC. In fact, the latter is only capable of producing resolution progressive bitstreams, which are SNR progressive within each resolution level only, whereas the former is capable of producing any mix of resolution, SNR,[3] component and position progressive bitstreams, in any order. Furthermore, the JPEG 2000 produced bitstreams are parseable and can be rather easily reorganized by a transcoder on the

---

[3] In JPEG 2000, SNR scalability is obtained by using layers, which are often associated with a SNR-based quality, but they are not restricted in any way and they can be associated with any other measure by the encoder, thus allowing for very flexible scalability.

Table 1
Functionality matrix[a]

	JPEG 2000	JPEG-LS	JPEG	MPEG-4 VTC	PNG
Lossless compression performance	+ + +	+ + + +	(+)	–	+ + +
Lossy compression performance	+ + + + +	+	+ + +	+ + + +	–
Progressive bitstreams	+ + + + +	–	+ +	+ + +	+
Region Of Interest (ROI) coding	+ + +	–	–	(+)	–
Arbitrary shaped objects	–	–	–	+ +	–
Random access	+ +	–	–	–	–
Low complexity	+ +	+ + + + +	+ + + + +	+	+ + +
Error resilience	+ + +	+ +	+ +	+ + +	+
Non-iterative rate control	+ + +	–	–	+	–
Genericity	+ + +	+ + +	+ +	+ +	+ + +

[a] A "+" indicates that it is supported, the more "+" the more efficiently or better it is supported. A "–" indicates that it is not supported. Parentheses indicate that a separate mode is required.

fly. Along the same lines, JPEG 2000 also provides random access (i.e. involving minimal decoding) to the block level in each subband, thus making possible to decode a region of the image without having to decode it as a whole.

Concerning error resilience JPEG 2000 offers higher protection than JPEG, as shown in Section 5. MPEG-4 VTC also offers error resilience features and although it could not be evaluated, [14] suggests that the results will be similar to JPEG 2000 ones. JPEG-LS does not offer any particular support for error resilience, besides restart markers, and has not been designed with it in mind. PNG offers error detection, but no concealment possibilities.

MPEG-4 VTC is the only evaluated standard which is capable of coding individual objects, of arbitrary shapes, independently. It is also capable of performing ROI coding, although it does so through tiling and thus requires that the ROIs be aligned on tile boundaries, which is quite restrictive. JPEG 2000 is the only algorithm which provides non-iterative rate control, in other words that is capable of producing a specified compression ratio without the need for an iterative scheme. MPEG-4 VTC can also be made to provide a limited form of non-iterative rate-control when bi-level quantization (BQ) is used.

All algorithms are able to compress different types of imagery across a wide range of bitrates, to which we refer to as *genericity*, in a quite efficient manner, although PNG has shown to adapt the best to particular image types (e.g. "target").

Overall, one can say that JPEG 2000 offers the richest set of features and provides superior rate-distortion performance. However, this comes at the price of additional complexity when compared to JPEG and JPEG-LS, which might be currently perceived as a disadvantage for some applications, as was the case for JPEG when it was first introduced.

7. Conclusions

This work aims at providing a comparative evaluation and assessment of JPEG 2000 performance, from various points of view, such as compression efficiency, complexity and set of supported functionality. The efficiency of various features that can be expected from a number of recent as well as most popular still image coding algorithms have been studied and compared. To do so, many aspects have been considered including genericity of the algorithm to code different types of data in lossless and lossy ways, and features such as error resilience, complexity, scalability, region of interest, embedded bitstream and so on.

The results presented in previous sections show that from a functionality point of view JPEG 2000 is a true improvement, providing lossy and lossless compression, progressive and parseable bitstreams, error resilience, region of interest, random access and other features in one integrated algorithm. However, while new standards provide higher compression efficiency there is no truly substantial improvement, in particular at medium and high-quality settings. This is especially true for lossy coding, even though the new standards, except for JPEG-LS, are significantly more complex than JPEG.

In any case, the choice of a standard for a particular application or product will depend on its requirements. In the cases where lossy compression is of interest and low complexity is of high priority, JPEG still provides a good solution. JPEG-LS stands out as the best option when only lossless compression is of interest, providing the best compression efficiency at a low complexity. PNG is also of interest in such cases, although the complexity of the encoder is much higher than that of JPEG-LS. As for MPEG-4 VTC, it appears to be of limited interest, except when the ability to code arbitrarily shaped objects is required. JPEG 2000 provides the most flexible solution, combining good compression performance with a rich set of features.

References

[1] A. Aydin Alatan, Minyi Zhao, A.N. Akansu, Unequal error protection of SPIHT encoded image bit streams, IEEE J. Selected Areas Commun. 18 (6) (June 2000) 814–818.

[2] M. Antonini, M. Barlaud, P. Mathieu, I. Daubechies, Image coding using wavelet transform, IEEE Trans. Image Process. 1 (2) (April 1992) 205–220.

[3] T. Chinen, A. Chien, Visual evaluation of JPEG 2000 color image compression performance, ISO/IEC JTC 1/SC 29/WG 1 N1583, March 2000, Tokyo, Japan.

[4] C. Christopoulos, J. Askelöf, M. Larsson, Efficient methods for encoding regions of interest in the upcoming JPEG 2000 still image coding standard, IEEE Signal Process. Lett. 7 (9) (2000) 247–249.

[5] T. Ebrahimi, C. Horne, MPEG-4 natural video coding – An overview, Signal Processing: Image Communication 15 (4–5) (January 2000) 365–385.

[6] ISO/IEC, ISO/IEC 14495-1:1999: information technology —lossless and near-lossless compression of continuous-tone still images: baseline, December 1999.

[7] ISO/IEC, ISO/IEC 14496-2:1999: information technology—coding of audio-visual objects—Part 2: visual, December 1999.

[8] ISO/IEC JTC 1/SC 29/WG 1, ISO/IEC FCD 15444-5: information technology—coding of still pictures—Part 5: reference software [WG 1 N 2020], January 2000.

[9] ISO/IEC JTC 1/SC 29/WG 1, ISO/IEC FDIS 15444-1: information technology—JPEG 2000 image coding system: core coding system [WG 1 N 1890], September 2000.

[10] ISO/IEC JTC1/SC29/WG1, Call for contributions for JPEG 2000 (JTC 1.29.14, 15444): image coding system [WG 1 N 505], March 1997, http://www.jpeg.org/public/wg1n505.pdf.

[11] D. Le Gall, A. Tabatai, Sub-band coding of digital images using symmetric short kernel filters and arithmetic coding techniques, in: Proceedings of the 1988 International Conference on Acoustics, Speech, and Signal Processing (ICASSP), April 1988, Vol. 2, pp. 761–764.

[12] I. Moccagatta, H. Chen, MPEG-4 visual texture coding: more than just compression, in: Proceedings of the International Conference on Consumer Electronics (ICCE) Chiba, Japan, June 1999, IEEE Press, New York, pp. 302–303.

[13] I. Moccagatta, M.Z. Coban, H. Chen, Wavelet-based image coding: comparison of mpeg-4 and jpeg-2000, in: Conference Record of the 33rd Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, October 1999, Vol. 2, pp. 1178–1182.

[14] I. Moccagatta, S. Soudagar, J. Liang, H. Chen, Error-resilient coding in JPEG 2000 and MPEG-4, IEEE J. Selected Areas Commun. 18 (6) (June 2000) 899–914.

[15] L. Nachtergaele, S. Wuytack, B. Vanhoof, P. Schelkens, J. Bormans, Study of the VM5.2 memory complexity for cost efficient implementation, ISO/IEC JTC1/SC29/WG1/N1469, Maui, Hawaii, USA, December 1999.

[16] A. Ortega, W. Jiang, P. Fernandez, C.G. Chrysafis, Implementations of the discrete wavelet transform: complexity, memory, and parallelization issues, in: Wavelet Applications in Signal and Image Processing VII, Proceedings of the SPIE, Denver, Co, USA, October 1999, Vol. 3813, pp. 386–400.

[17] W.B. Pennebaker, J.L. Mitchell, JPEG: Still Image Data Compression Standard, Van Nostrand Reinhold, New York, 1992.

[18] F. Pereira, MPEG-4: Why, what, how and when?, Signal Processing: Image Communication 15 (4–5) (2000) 271–279.

[19] A. Said, W.A. Pearlman, Reversible image compression via multiresolution representation and predictive coding, in: Visual Communications and Image Processing, Proceedings of the SPIE, Cambridge, MA, USA, November 1993, Vol. 2094, pp. 664–674.

[20] A. Said, W.A. Pearlman, A new and efficient image codec based on set partitioning in hierarchical trees, IEEE Trans. Circuits Systems Video Technol. 6 (3) (June 1996) 243–250.

[21] D. Santa-Cruz, T. Ebrahimi, An analytical study of JPEG 2000 functionalities, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), Vancouver, Canada, September 2000, Vol. 2, pp. 49–52.

[22] D. Santa-Cruz, T. Ebrahimi, J. Askelöf, M. Larsson, C. Christopoulos, JPEG 2000 still image coding versus other standards, in: SPIE's 45th Annual Meeting, Applications of Digital Image Processing XXIII, Proceedings of SPIE, San Diego, California, August 2000, Vol. 4115, pp. 446–454.

[23] I. Soudagar, H.-J. Lee, P. Hatrack, Y.-Q. Zhang, Scalable wavelet coding for synthetic/natural hybrid images, IEEE Trans. Circuits Systems Video Technol. 9 (2) (March 1999) 244–254.

[24] D. Taubman, High performance scalable image compression with EBCOT, IEEE Trans. Image Process. 9 (7) (July 2000) 1158–1170.

[25] W3C, PNG (Portable Network Graphics) Specification, October 1996, http://www.w3.org/TR/REC-png.

[26] G.K. Wallace, The JPEG still picture compression standard, in: IEEE Transactions on Consumer Electronics, February 1992, Vol. 38, pp. xviii–xxxiv.

[27] M.J. Weinberger, G. Seroussi, G. Sapiro, The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS, IEEE Trans. Image Process. 9 (8) (August 2000) 1309–1324.

[28] X. Wu, N. Memon, Context-based, adaptive, lossless image coding, IEEE Trans. Commun. 45 (4) (April 1997) 437–444.