

The JPEG 2000 Still Image Compression Standard

Athanassios Skodras, Charilaos Christopoulos, and Touradj Ebrahimi

The development of standards (emerging and established) by the International Organization for Standardization (ISO), the International Telecommunications Union (ITU), and the International Electrotechnical Commission (IEC) for audio, image, and video, for both transmission and storage, has led to worldwide activity in developing hardware and software systems and products applicable to a number of diverse disciplines [7], [22], [23], [55], [56], [73]. Although the standards implicitly address the basic encoding operations, there is freedom and flexibility in the actual design and development of devices. This is because only the syntax and semantics of the bit stream for decoding are specified by standards, their main objective being the compatibility and interoperability among the systems (hardware/software) manufactured by different companies. There is, thus, much room for innovation and ingenuity.

Since the mid 1980s, members from both the ITU and the ISO have been working together to establish a joint international standard for the compression of grayscale and color still images. This effort has been known as JPEG, the Joint Photographic Experts Group. (The “joint” in JPEG refers to the collaboration between ITU and ISO; see Fig. 1). Officially, JPEG corresponds to the ISO/IEC international standard 10928-1, digital compression and coding of continuous-tone (multilevel) still images or to the ITU-T Recommendation T.81. The text in both these ISO and ITU-T documents is identical. The process was such that, after evaluating a number of coding schemes, the JPEG members selected a discrete cosine transform (DCT)-based method in 1988. From 1988 to 1990, the JPEG group continued its work by simulating, testing and documenting the algorithm. JPEG became Draft International Standard (DIS) in 1991 and International Standard (IS) in 1992 [55], [73].

With the continual expansion of multimedia and Internet applications, the needs and requirements of the technologies used grew and evolved. In March 1997, a

new call for contributions was launched for the development of a new standard for the compression of still images, the JPEG 2000 standard [28], [29]. This project, JTC 1.29.14 (ISO/IEC 15444-1 or ITU-T Rec. T.800), was intended to create a new image coding system for different types of still images (bilevel, gray level, color, multicomponent), with different characteristics (natural images, scientific, medical, remote sensing, text, rendered graphics, etc.) allowing different imaging models (client/server, real-time transmission, image library archival, limited buffer and bandwidth resources, etc.) preferably within a unified system. This coding system should provide low bit-rate operation with rate distortion and subjective image quality performance superior to existing standards, without sacrificing performance at other points in the rate-distortion spectrum, and at the same time incorporating many interesting features. One

of the aims of the standardization committee has been the development of Part I, which could be used on a royalty- and fee-free basis. This is important for the standard to become widely accepted, in the same manner as the original JPEG with Huffman coding is now. The standardization process, which is coordinated by the JTC1/SC29/WG1 of ISO/IEC (Fig. 1) has already (since December 2000) produced the International Standard (IS) for Part I [41].

In this article the structure of Part I of the JPEG 2000 standard is presented and performance comparisons with established standards are reported. This article is intended to serve as a tutorial for the JPEG 2000 standard. In the next section the main application areas and

their requirements are given. The architecture of the standard follows afterwards, with the description of the tiling, multicomponent transformations, wavelet transforms, quantization and entropy coding. Some of the most significant features of the standard are presented next, such as region-of-interest coding, scalability, visual weighting, error resilience and file format aspects. Finally, some com-



© 2001 IMAGESTATE

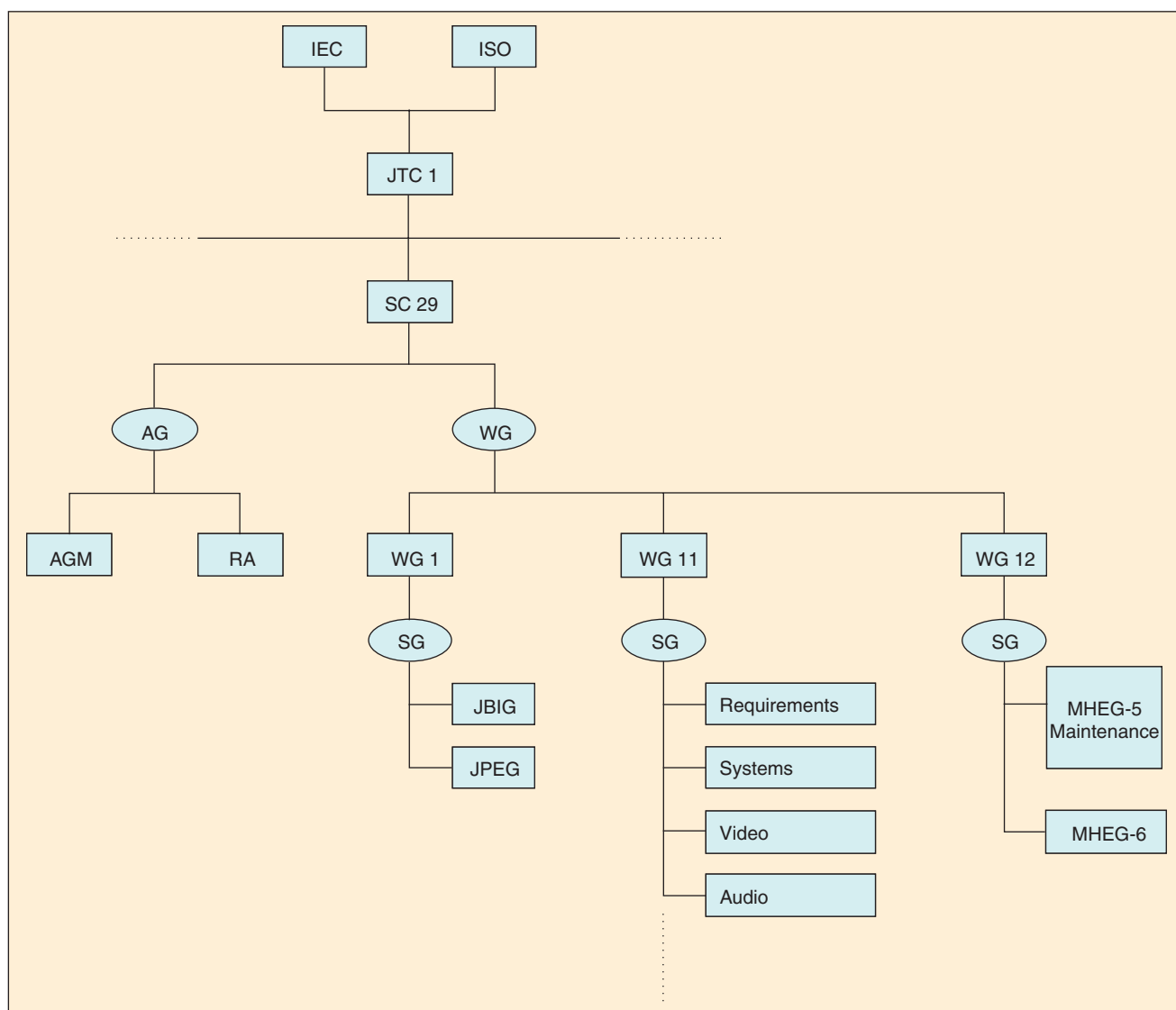
parative results are reported and the future parts of the standard are discussed.

Why Another Still Image Compression Standard?

The JPEG standard has been in use for almost a decade now. It has proved a valuable tool during all these years, but it cannot fulfill the advanced requirements of today. Today's digital imagery is extremely demanding, not only from the quality point of view, but also from the image size aspect. Current image size covers orders of magnitude, ranging from web logos of size of less than 100 Kbits to high quality scanned images of approximate size of 40 Gbits [20], [33], [43], [48]. The JPEG 2000 international standard represents advances in image compression

technology where the image coding system is optimized not only for efficiency, but also for scalability and interoperability in network and mobile environments. Digital imaging has become an integral part of the Internet, and JPEG 2000 is a powerful new tool that provides power capabilities for designers and users of networked image applications [41].

The JPEG 2000 standard provides a set of features that are of importance to many high-end and emerging applications by taking advantage of new technologies. It addresses areas where current standards fail to produce the best quality or performance and provides capabilities to markets that currently do not use compression. The markets and applications better served by the JPEG 2000 standard are Internet, color facsimile, printing, scanning (consumer and prepress), digital photography, remote



▲ 1. Structure of the standardization bodies and related terminology. ISO: International Organization for Standardization, ITU: International Telecommunications Union, IEC: International Electrotechnical Commission, JTC: Joint Technical Committee on Information Technology, SC: Subcommittee, SG: study group, WG: working group, AHG: ad hoc group, JPEG: Joint Photographic Experts Group, JBIG: Joint Bi-Level Image Group, MPEG: Moving Picture Experts Group, MHEG: Multimedia Hypermedia Experts Group, AG: advisory group, AGM: Advisory Group on Management, and RA: registration authority.

JPEG 2000 represents advances in image compression technology where the image coding system is optimized for efficiency, scalability, and interoperability in network and mobile environments.

sensing, mobile, medical imagery, digital libraries/archives, and E-commerce. Each application area imposes some requirements that the standard, up to a certain degree, should fulfill. Some of the most important features that this standard should possess are the following [33]:

▲ *Superior low bit-rate performance:* This standard should offer performance superior to the current standards at low bit rates (e.g., below 0.25 b/p for highly detailed gray-scale images). This significantly improved low bit-rate performance should be achieved without sacrificing performance on the rest of the rate-distortion spectrum. Network image transmission and remote sensing are some of the applications that need this feature.

▲ *Continuous-tone and bilevel compression:* It is desired to have a coding standard that is capable of compressing both continuous-tone and bilevel images. If feasible, this standard should strive to achieve this with similar system resources. The system should compress and decompress images with various dynamic ranges (e.g., 1 to 16 bits) for each color component. Examples of applications that can use this feature include compound documents with images and text, medical images with annotation overlays, and graphic and computer generated images with binary and near to binary regions, alpha and transparency planes, and facsimile.

▲ *Lossless and lossy compression:* It is desired to provide lossless compression naturally in the course of progressive decoding. Examples of applications that can use this

feature include medical images, where loss is not always tolerated; image archival applications, where the highest quality is vital for preservation but not necessary for display; network applications that supply devices with different capabilities and resources; and prepress imagery. It is also desired that the standard should have the property of creating embedded bit stream and allow progressive lossy to lossless buildup.

▲ *Progressive transmission by pixel accuracy and resolution:* Progressive transmission that allows images to be reconstructed with increasing pixel accuracy or spatial resolution is essential for many applications such as web browsing, image archival and printing.

▲ *Region-of-interest (ROI) coding:* Often there are parts of an image that are of greater importance than others. This feature allows users to define certain ROIs in the image to be coded and transmitted in a better quality and less distortion than the rest of the image.

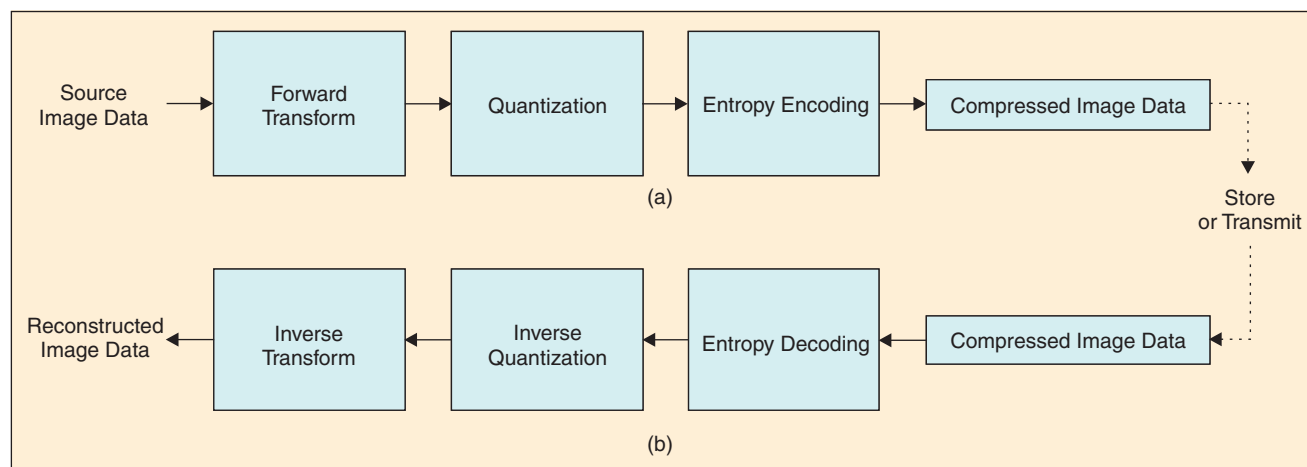
▲ *Open architecture:* It is desirable to allow open architecture to optimize the system for different image types and applications. With this feature, a decoder is only required to implement the core tool set and the parser that understands the code stream.

▲ *Robustness to bit errors:* It is desirable to consider robustness to bit errors while designing the code stream. One application, where this is important, is transmission over wireless communication channels. Portions of the code stream may be more important than others in determining decoded image quality. Proper design of the code stream can aid subsequent error correction systems in alleviating catastrophic decoding failures.

▲ *Protective image security:* Protection of a digital image can be achieved by means of different approaches such as watermarking, labeling, stamping, or encryption. JPEG 2000 image files should have provisions for such possibilities.

The JPEG 2000 Compression Engine

The JPEG 2000 compression engine (encoder and decoder) is illustrated in block diagram form in Fig. 2. At



▲ 2. General block diagram of the JPEG 2000 (a) encoder and (b) decoder.

the encoder, the discrete transform is first applied on the source image data. The transform coefficients are then quantized and entropy coded before forming the output code stream (bit stream). The decoder is the reverse of the encoder. The code stream is first entropy decoded, dequantized, and inverse discrete transformed, thus resulting in the reconstructed image data. Although this general block diagram looks like the one for the conventional JPEG, there are radical differences in all of the processes of each block of the diagram. A quick overview of the whole system is as follows:

- ▲ The source image is decomposed into components.
- ▲ The image components are (optionally) decomposed into rectangular tiles. The tile-component is the basic unit of the original or reconstructed image.
- ▲ A wavelet transform is applied on each tile. The tile is decomposed into different resolution levels.
- ▲ The decomposition levels are made up of subbands of coefficients that describe the frequency characteristics of local areas of the tile components, rather than across the entire image component.
- ▲ The subbands of coefficients are quantized and collected into rectangular arrays of “code blocks.”
- ▲ The bit planes of the coefficients in a code block (i.e., the bits of equal significance across the coefficients in a code block) are entropy coded.
- ▲ The encoding can be done in such a way that certain regions of interest can be coded at a higher quality than the background.
- ▲ Markers are added to the bit stream to allow for error resilience.
- ▲ The code stream has a main header at the beginning that describes the original image and the various decomposition and coding styles that are used to locate, extract, decode and reconstruct the image with the desired resolution, fidelity, region of interest or other characteristics.

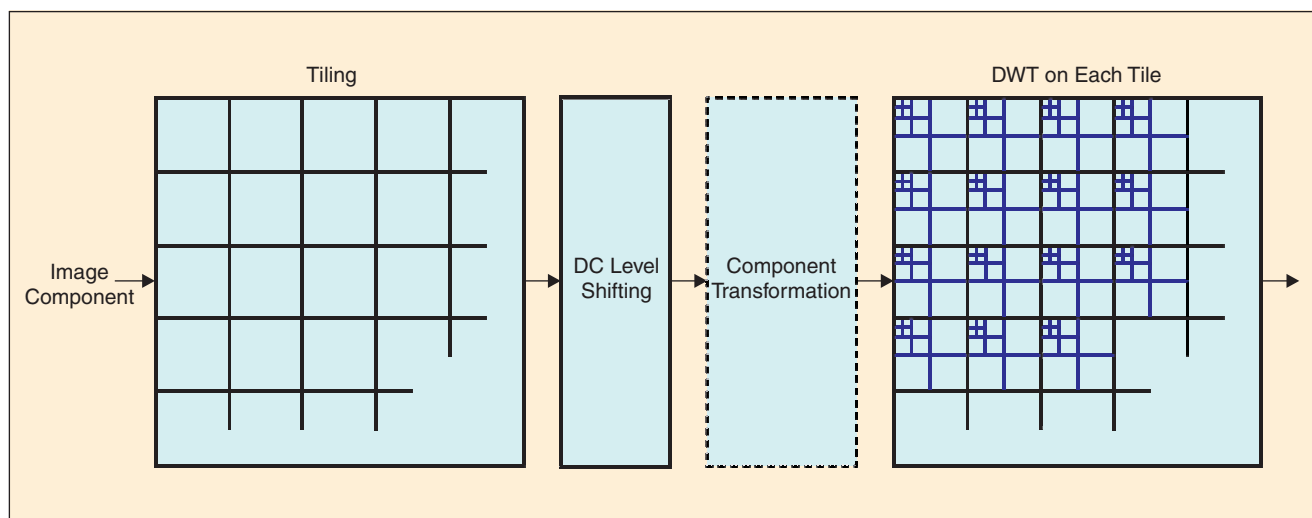
For the clarity of presentation we have decomposed the whole compression engine into three parts: the pre-

processing, the core processing, and the bit-stream formation part, although there exist high inter-relation between them. In the preprocessing part the image tiling, the dc-level shifting and the component transformations are included. The core processing part consists of the discrete transform, the quantization and the entropy coding processes. Finally, the concepts of the precincts, code blocks, layers, and packets are included in the bit-stream formation part.

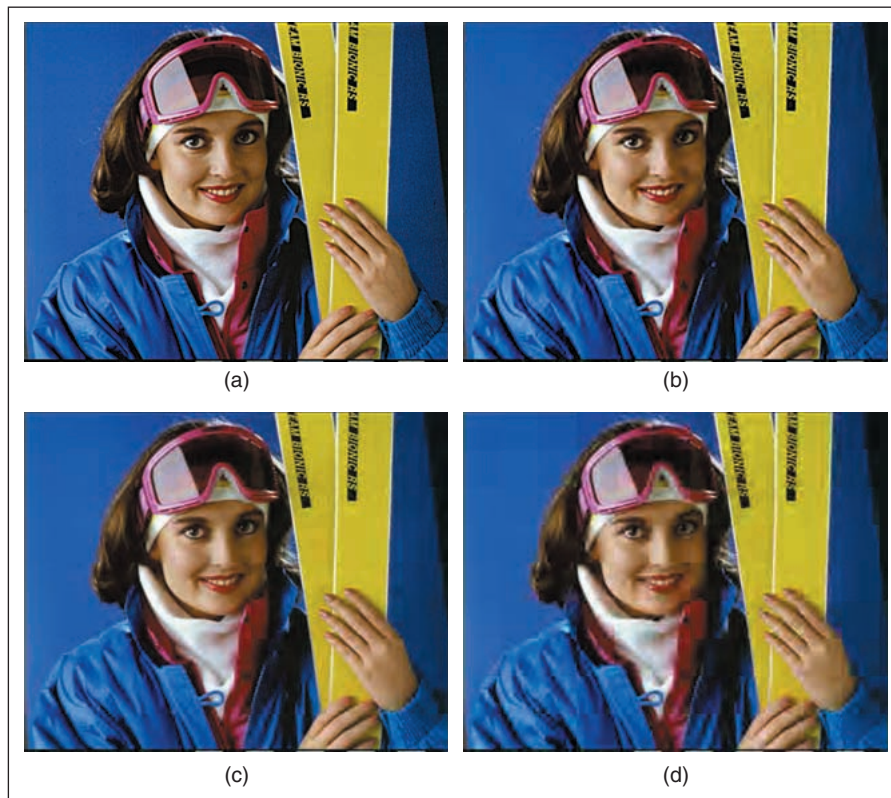
Preprocessing

Image Tiling

The term “tiling” refers to the partition of the original (source) image into rectangular nonoverlapping blocks (tiles), which are compressed independently, as though they were entirely distinct images [13], [24], [38], [39], [48], [63]. All operations, including component mixing, wavelet transform, quantization and entropy coding are performed independently on the image tiles (Fig. 3). The tile component is the basic unit of the original or reconstructed image. Tiling reduces memory requirements, and since they are also reconstructed independently, they can be used for decoding specific parts of the image instead of the whole image. All tiles have exactly the same dimensions, except maybe those at the boundary of the image. Arbitrary tile sizes are allowed, up to and including the entire image (i.e., the whole image is regarded as one tile). Components with different subsampling factors are tiled with respect to a high-resolution grid, which ensures spatial consistency on the resulting tile components. As expected, tiling affects the image quality both subjectively (Fig. 4) and objectively (Table 1). Smaller tiles create more tiling artifacts compared to larger tiles (PSNR values are the average over all components). In other words, larger tiles perform visually better than smaller tiles. Image degradation is more severe in the case of low bit rate than the case of high bit rate. It is seen, for example, that at 0.125 b/p there is a quality difference of more than 4.5 dB be-



▲ 3. Tiling, dc-level shifting, color transformation (optional) and DWT of each image component.



▲ 4. Image :“ski” of size 720×576 (courtesy of Phillips Research, UK): (a) original image, (b)-(d) reconstructed images after JPEG 2000 compression at 0.25 bpp: (b) without tiling, (c) with 128×128 tiling, and (d) with 64×64 tiling.

tween no-tiling and tiling at 64×64 , while at 0.5 b/p this difference is reduced to approximately 1.5 dB.

DC Level Shifting

Prior to computation of the forward discrete wavelet transform (DWT) on each image tile, all samples of the image tile component are dc level shifted by subtracting the same quantity 2^{P-1} , where P is the component's precision. DC level shifting is performed on samples of components that are unsigned only. Level shifting does not affect variances. It actually converts an unsigned representation to a two's complement representation, or vice versa [55], [56]. If color transformation is used, dc level shifting is performed prior to the computation of the forward

ward component transform (Figs. 3 and 5). At the decoder side, inverse dc level shifting is performed on reconstructed samples by adding to them the bias 2^{P-1} after the computation of the inverse component transform.

Component Transformations

JPEG 2000 supports multiple-component images. Different components need not have the same bit depths nor need to all be signed or unsigned [38], [39]. For reversible (i.e., lossless) systems, the only requirement is that the bit depth of each output image component must be identical to the bit depth of the corresponding input image component.

Component transformations improve compression and allow for visually relevant quantization. The standard supports two different component transformations, one irreversible component transformation (ICT) that can be used for lossy coding and one reversible

component transformation (RCT) that may be used for lossless or lossy coding, and all this in addition to encoding without color transformation. The block diagram of the JPEG 2000 multicomponent encoder is depicted in Fig. 5. (Without restricting the generality, only three components are shown in the figure. These components could correspond to the RGB of a color image.)

Since the ICT may only be used for lossy coding, it may only be used with the 9/7 irreversible wavelet transform. (See also next section.) The forward and the inverse ICT transformations are achieved by means of (1a) and (1b), respectively [7], [38], [56]

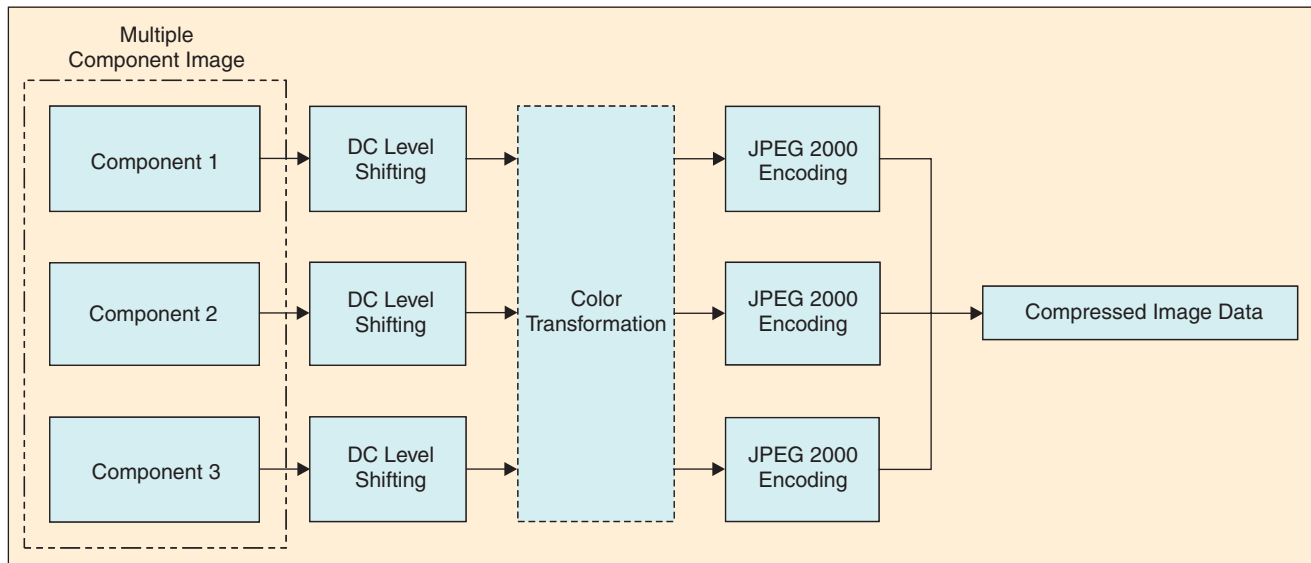
$$\begin{pmatrix} \gamma \\ C_b \\ C_r \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.16875 & -0.33126 & 0.5 \\ 0.5 & -0.41869 & -0.08131 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad (1a)$$

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1.0 & 0 & 1.402 \\ 1.0 & -0.34413 & -0.71414 \\ 1.0 & 1.772 & 0 \end{pmatrix} \begin{pmatrix} \gamma \\ C_b \\ C_r \end{pmatrix} \quad (1b)$$

Since the RCT may be used for lossless or lossy coding, it may only be used with the 5/3 reversible wavelet transform. (See also next section.) The RCT is a decorrelating transformation, which is applied to the three first components of an image. Three goals are achieved by this transformation, namely, color decorrelation for efficient

Table 1. The Effect of Tiling on Image Quality.

Tiling	No Tiling	Tiles of Size 128×128	Tiles of Size 64×64
Bit Rate (b/p)			
0.125	24.75	23.42	20.07
0.25	26.49	25.69	23.95
0.5	28.27	27.79	26.80
PSNR (in dB) for the color image “ski” (of size 720×576 pixels per component)			



▲ 5. The JPEG 2000 multiple component encoder. Color transformation is optional. If employed, it can be irreversible or reversible.

compression, reasonable color space with respect to the human visual system for quantization, and ability of having lossless compression, i.e., exact reconstruction with finite integer precision. For the RGB components, the RCT can be seen as an approximation of a YUV transformation. All three of the components shall have the same sampling parameters and the same bit depth. There shall be at least three components if this transform is used. The forward and inverse RCT is performed by means of (2a) and (2b), respectively, where the subscript r stands for reversible

$$\begin{pmatrix} \Upsilon_r \\ V_r \\ U_r \end{pmatrix} = \begin{pmatrix} \left\lfloor \frac{R+2G+B}{4} \right\rfloor \\ R-G \\ B-G \end{pmatrix} \quad (2a)$$

$$\begin{pmatrix} G \\ R \\ B \end{pmatrix} = \begin{pmatrix} \Upsilon_r - \left\lfloor \frac{U_r + V_r}{4} \right\rfloor \\ V_r + G \\ U_r + G \end{pmatrix} \quad (2b)$$

where $\lfloor a \rfloor$ is the largest integer not exceeding a .

A subjective quality evaluation of the different color spaces can be found in [37] and [50]. Performance comparisons between lossless compression (i.e., using RCT and the 5/3 filter) and decompression at a certain bitrate, and lossy compression (i.e., using ICT and the 9/7 filter) and decompression at the same bit rate, has shown that the later produces substantially better results, as shown in Table 2.

An effective way to reduce the amount of data in JPEG is to use an RGB to YCrCb decorrelation transform followed by subsampling of the chrominance (C_r, C_b) components. This is not recommended for use in JPEG 2000,

since the multiresolution nature of the wavelet transform may be used to achieve the same effect. For example, if the HL, LH, and HH subbands of a component's wavelet decomposition are discarded and all other subbands retained, a 2:1 subsampling is achieved in the horizontal and vertical dimensions of the component.

Core Processing

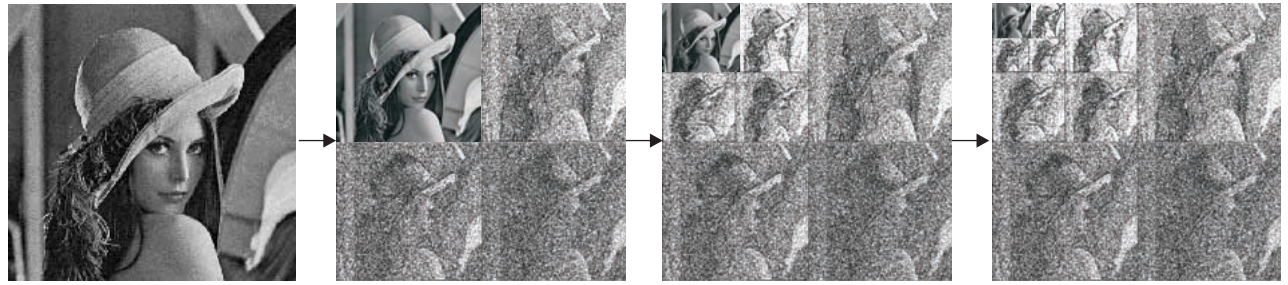
Wavelet Transform

Wavelet transform is used for the analysis of the tile components into different decomposition levels [9], [26], [70], [71]. These decomposition levels contain a number of subbands, which consist of coefficients that describe the horizontal and vertical spatial frequency characteristics of the original tile component. In Part I of the JPEG 2000 standard only power of 2 decompositions are allowed in the form of dyadic decomposition as shown in Fig. 6 for the image "Lena".

To perform the forward DWT the standard uses a one-dimensional (1-D) subband decomposition of a 1-D set of samples into low-pass and high-pass samples. Low-pass samples represent a down-sampled, low-resolution version of the original set. High-pass samples rep-

Table 2. The effect of component transformation on the compression efficiency for the ski image. RCT is employed in the lossless case and ICT in the lossy case. No tiling is used.

	Without Color Transformation	With Color Transformation
Lossless compression	16.88 b/p	14.78 b/p
Lossy compression at 0.25 b/p	25.67 dB	26.49 dB



▲ 6. Three-level dyadic wavelet decomposition of the image "Lena."

Table 3. Daubechies 9/7 Analysis and Synthesis Filter Coefficients.

Analysis Filter Coefficients		
i	Low-Pass Filter $h_L(i)$	High-Pass Filter $h_H(i)$
0	0.6029490182363579	1.115087052456994
± 1	0.2668641184428723	-0.5912717631142470
± 2	-0.07822326652898785	-0.05754352622849957
± 3	-0.01686411844287495	0.09127176311424948
± 4	0.02674875741080976	
Synthesis Filter Coefficients		
i	Low-Pass Filter $g_L(i)$	High-Pass Filter $g_H(i)$
0	1.115087052456994	0.6029490182363579
± 1	0.5912717631142470	-0.2668641184428723
± 2	-0.05754352622849957	-0.07822326652898785
± 3	-0.09127176311424948	0.01686411844287495
± 4		0.02674875741080976

Table 4. Le Gall 5/3 Analysis and Synthesis Filter Coefficients.

	Analysis Filter Coefficients		Synthesis Filter Coefficients	
	Low-Pass Filter $h_L(i)$	High-Pass Filter $h_H(i)$	Low-Pass Filter $g_L(i)$	High-Pass Filter $g_H(i)$
0	6/8	1	1	6/8
± 1	2/8	-1/2	1/2	-2/8
± 2	-1/8			-1/8

represent a down-sampled residual version of the original set, needed for the perfect reconstruction of the original set from the low-pass set. The DWT can be irreversible or reversible. The default irreversible transform is implemented by means of the Daubechies 9-tap/7-tap filter [4]. The analysis and the corresponding synthesis filter coefficients are given in Table 3. The default reversible transformation is implemented by means of the Le Gall 5-tap/3-tap filter, the coefficients of which are given in Table 4 [2], [12], [46].

The standard can support two filtering modes: convolution based and lifting based. For both modes to be implemented, the signal should first be extended periodically as has been demonstrated in the previous article by Usevitch [70] (Fig. 7). This periodic symmetric extension is used to ensure that for the filtering operations that take place at both boundaries of the signal, one signal sample exists and spatially corresponds to each coefficient of the filter mask. The number of additional samples required at the boundaries of the signal is therefore filter-length dependent. The symmetric extension of the boundary is of type (1,1), i.e., the first and the last samples appear only once and are whole sample (WS) since the length of the kernel is odd [3], [10], [11].

Convolution-based filtering consists in performing a series of dot products between the two filter masks and the extended 1-D signal. Lifting-based filtering consists of a sequence of very simple filtering operations for which alternately odd sample values of the signal are updated with a weighted sum of even sample values, and even sample values are updated with a weighted sum of odd sample values (Fig. 8) [3], [45], [62], [64], [65]. For the reversible (lossless) case the results are rounded to integer values. The lifting-based filtering [2], [38] for the 5/3 analysis filter is achieved by means of

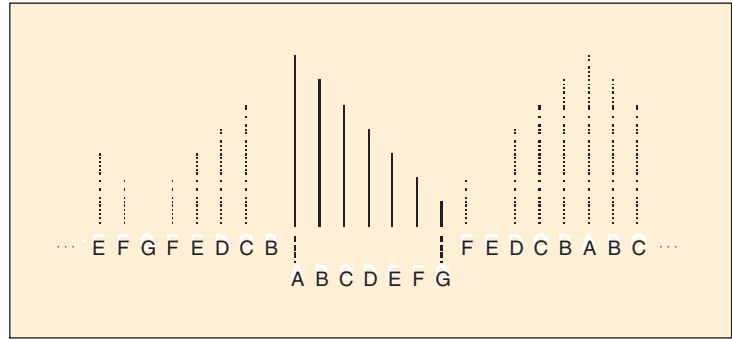
$$y(2n+1) = x_{\text{ext}}(2n+1) - \left\lfloor \frac{x_{\text{ext}}(2n) + x_{\text{ext}}(2n+2)}{2} \right\rfloor \quad (3a)$$

$$y(2n) = x_{\text{ext}}(2n) + \left\lfloor \frac{y(2n-1) + y(2n+1) + 2}{4} \right\rfloor \quad (3b)$$

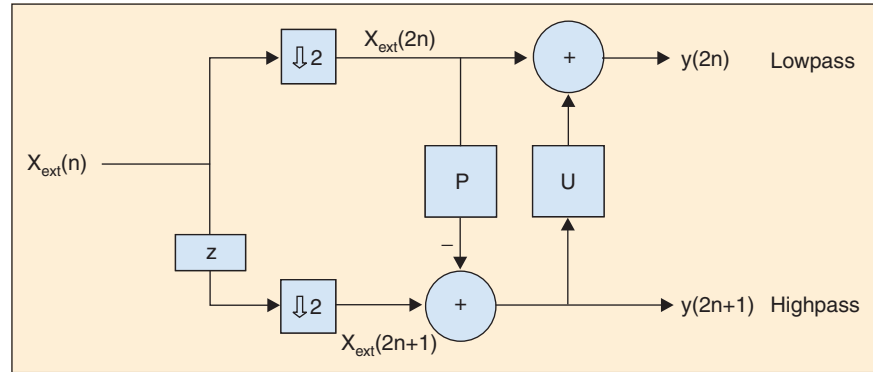
where x_{ext} is the extended input signal and y is the output signal. The 5/3 filter allows repetitive encoding and de-

coding of an image without any loss. Of course, this is true when the decompressed image values are not clipped when they fall outside the full dynamic range (i.e., 0-255 for an 8 b/p image) [48].

Traditional wavelet transform implementations require the whole image to be buffered and the filtering operation to be performed in vertical and horizontal directions. While filtering in the horizontal direction is very simple, filtering in the vertical direction is more cumbersome. Filtering along a row requires one row to be read; filtering along a column requires the whole image to be read. The line-based wavelet transform overcomes this difficulty, providing exactly the same transform coefficients as the traditional wavelet transform implementation [17], [18], [38]. However, the line-based wavelet transform alone does not provide a complete line-based encoding paradigm for JPEG 2000. A complete row-based coder has to take also into account all the subsequent coding stages up to the entropy coding. Such an algorithm is described in [17] and [18].



▲ 7. Periodic symmetric extension of the finite length signal "ABCDEFGG."



▲ 8. The forward (analysis) wavelet transform using lifting. P and U stand for prediction and update, respectively.

Quantization

After transformation, all coefficients are quantized. Uniform scalar quantization with dead-zone about the origin is used in Part I and trellis coded quantization (TCQ) in Part II of the standard [38], [39], [44]. Quantization is the process by which the coefficients are reduced in precision. This operation is lossy, unless the quantization step is 1 and the coefficients are integers, as produced by the reversible integer 5/3 wavelet. Each of the transform coefficients $a_b(u, v)$ of the subband b is quantized to the value $q_b(u, v)$ according to the formula [38], [39]

$$q_b(u, v) = \text{sign}(a_b(u, v)) \left\lceil \frac{|a_b(u, v)|}{\Delta_b} \right\rceil. \quad (4)$$

The quantization step-size Δ_b is represented relative to the dynamic range of subband b . In other words, the JPEG 2000 standard supports separate quantization step-sizes for each subband [70]. However, one quantization step-size is allowed per subband. The dynamic range depends on the number of bits used to represent the original image tile component and on the choice of the wavelet transform. All quantized transform coefficients are signed values even when the original components are unsigned. These coefficients are expressed in a sign-magnitude representation prior to coding. For reversible compression, the quantization step-size is required to be one.

Entropy Coding

Entropy coding is achieved by means of an arithmetic coding system that compresses binary symbols relative to an adaptive probability model associated with each of 18 different coding contexts. The MQ coding algorithm is used to perform this task and to manage the adaptation of the conditional probability models [39], [55]. This algorithm has been selected in part for compatibility reasons with the arithmetic coding engine used by the JBIG2 compression standard and every effort has been made to ensure commonality between implementations and surrounding intellectual property issues for JBIG2 and JPEG 2000 [39]. The recursive probability interval subdivision of Elias coding is the basis for the binary *arithmetic coding* process. With each binary decision, the current probability interval is subdivided into two subintervals, and the code stream is modified (if necessary) so that it points to the base (the lower bound) of the probability subinterval assigned to the symbol, which occurred. Since the coding process involves addition of binary fractions rather than concatenation of integer code words, the more probable binary decisions can often be coded at a cost of much less than one bit per decision [7], [55], [56].

As mentioned above, JPEG 2000 uses a very restricted number of contexts for any given type of bit. This allows rapid probability adaptation and decreases the cost of independently coded segments. The context models are always reinitialized at the beginning of each code block and the arithmetic coder is always terminated

at the end of each block (i.e., once, at the end of the last subbit plane). This is useful for error resilience also. (A code block is the fundamental entity for entropy coding—see also next section.)

In addition to the above, a lazy coding mode is used to reduce the number of symbols that are arithmetically coded [39], [48]. According to this mode, after the fourth bit plane is coded, the first and second pass are included as raw (uncompressed data), i.e., the MQ coder is bypassed, while only the third coding pass of each bit plane employs arithmetic coding. This results in significant speedup for software implementations at high bit rates. Lazy coding has a negligible effect on compression efficiency for most natural images, but not for compound imagery [69].

Bit-Stream Formation

Precincts and code blocks

After quantization, each subband is divided into rectangular blocks, i.e., nonoverlapping rectangles. Three spatially consistent rectangles (one from each subband at each resolution level) comprise a packet partition location or precinct. Each precinct is further divided into nonoverlapping

rectangles, called code blocks, which form the input to the entropy coder (Fig. 9). The size of the code block is typically 64×64 and no less than 32×32 .

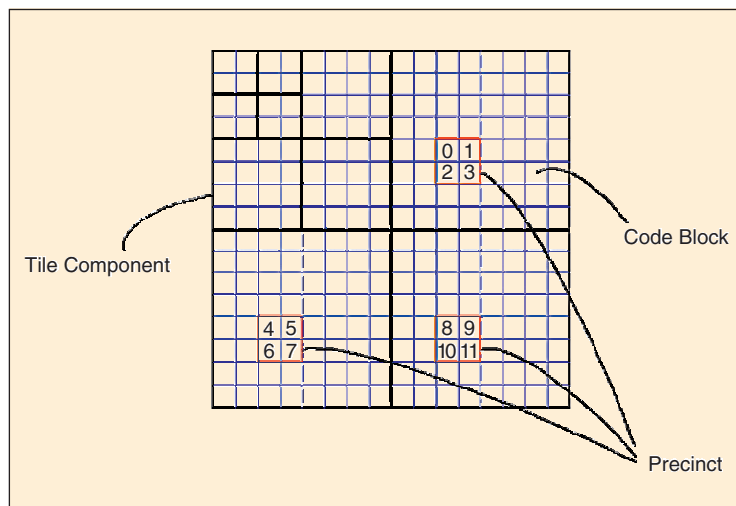
Within each subband the code blocks are visited in raster order. These code blocks are then coded a bit plane at a time starting with the most significant bit plane with a nonzero element to the least significant bit plane [67]–[69]. Each code block is coded entirely independently, without reference to other blocks in the same or other subbands, something that is in contrary to the approach adopted by the zero-tree coder described by Uevitch [70]. This independent embedded block coding offers significant benefits, such as spatial random access to the image content, efficient geometric manipulations, error resilience, parallel computations during coding or decoding, etc. The individual bit planes of the coefficients in a code block are coded within three coding passes. Each of these coding passes collects contextual information about the bit plane data. The arithmetic coder uses this contextual information and its internal state to generate a compressed bit stream. Different termination mechanisms allow different levels of independent extraction of this coding pass data.

Each bit plane of a code block is scanned in a particular order (Fig. 10). Starting from the top left, the first four bits of the first column are scanned. Then the first four bits of the second column, until the width of the code block is covered. Then the second four bits of the first column are scanned and so on. A similar vertical scan is continued for any leftover rows on the lowest code blocks in the subband [38]. This stripe height of 4 has been carefully selected to facilitate efficient hardware and software implementations [68], [69].

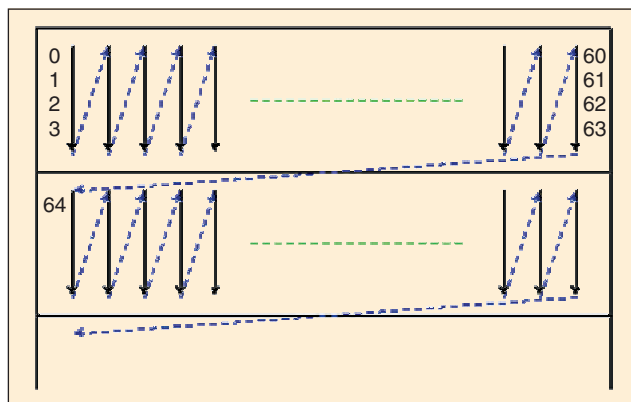
Each coefficient bit in the bit plane is coded in only one of the three coding passes, namely the significance propagation, the magnitude refinement, and the cleanup pass. For each pass, contexts are created which are provided to the arithmetic coder [48], [69].

During the significance propagation pass, a bit is coded if its location is not significant, but at least one of its eight-connect neighbors is significant. Nine context bins are created based on how many and which ones are significant. If a coefficient is significant then it is given a value of 1 for the creation of the context, otherwise it is given a value of 0. The mapping of the contexts also depends on which subband (at a given decomposition level) the code block is in. The significance propagation pass includes only bits of coefficients that were insignificant (the significance bit has yet to be encountered) and have a nonzero context. All other coefficients are skipped. The context is delivered to the arithmetic decoder (along with the bit stream) and the decoded coefficient bit is returned.

The second pass is the magnitude refinement pass. During this pass, all bits that became significant in a pre-



▲ 9. Partition of a tile component into code blocks and precincts.



▲ 10. Scan pattern of each bit plane of each code block.

vious biplane are coded. The magnitude refinement pass includes the bits from coefficients that are already significant (except those that have just become significant in the immediately preceding significance propagation pass). The context used is determined by the summation of the significance state of the horizontal, vertical, and diagonal neighbors. These are the states as currently known to the decoder, not the states used before the significance decoding pass. Further, it is dependent on whether this is the first refinement bit (the bit immediately after the significance and sign bits) or not.

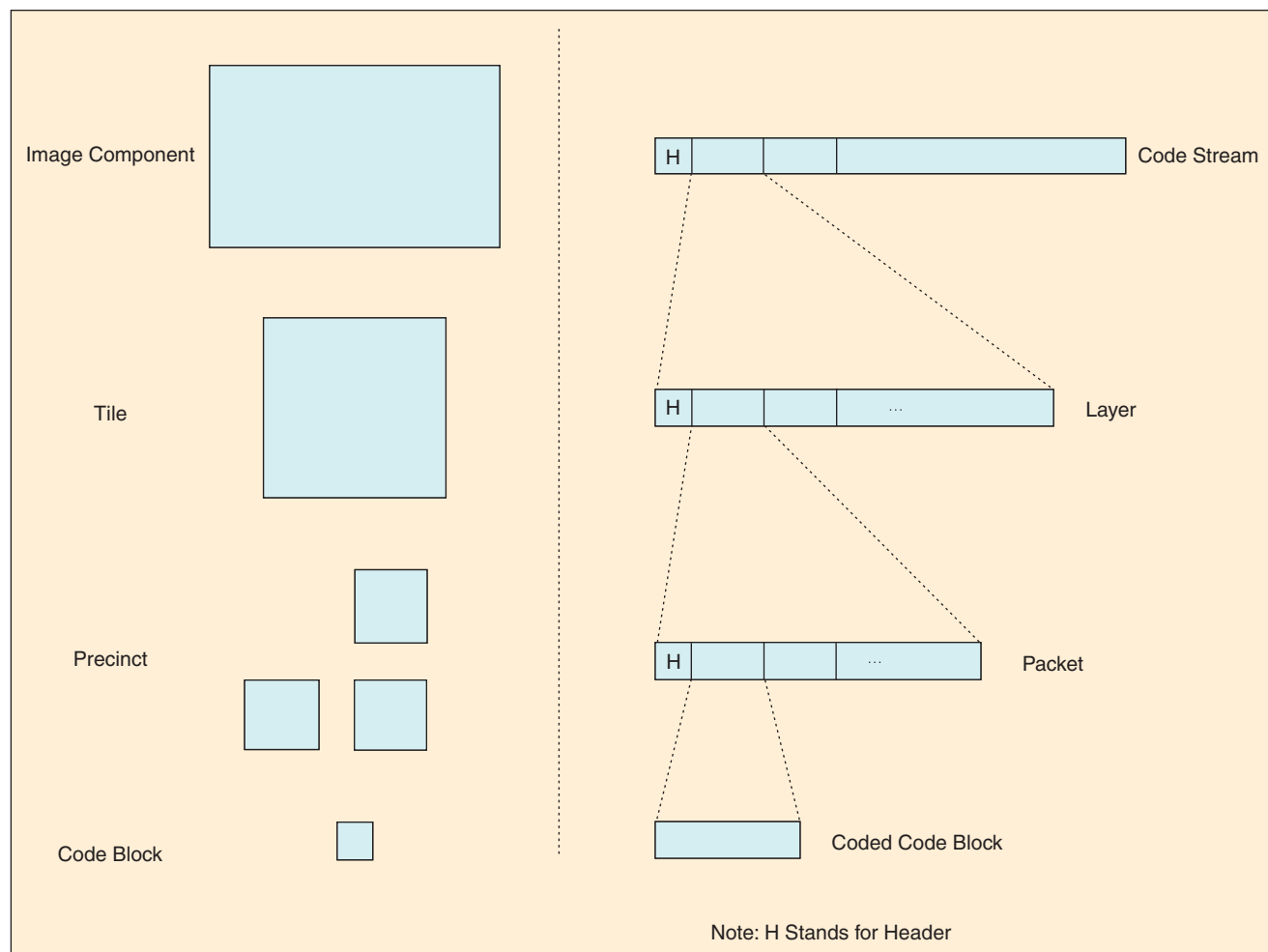
The final pass is the clean-up pass in which all bits not encoded during the previous passes are encoded (i.e., coefficients that are insignificant and had the context value of zero during the significance propagation pass). The cleanup pass not only uses the neighbor context, like that of the significance propagation pass, but also a run-length context. Run coding occurs when all four locations in the column of the scan are insignificant and each has only insignificant neighbors [38], [39], [48], [69].

Packets and Layers

For each code block, a separate bit stream is generated. No information from other blocks is utilized during the genera-

Comparative results have shown that JPEG 2000 is indeed superior to established still image compression standards.

tion of the bit stream for a particular block. Rate distortion optimization is used to allocate truncation points to each code block. The bit stream has the property that it can be truncated to a variety of discrete lengths, and the distortion incurred, when reconstructing from each of these truncated subsets, is estimated and denoted by the mean squared error. During the encoding process, the lengths and the distortions are computed and temporarily stored with the compressed bit stream itself. The compressed bit streams from each code block in a precinct comprise the body of a packet. A collection of packets, one from each precinct of each resolution level, comprises the layer (Fig. 11). A packet could be interpreted as one quality increment for one resolution level at one spatial location, since precincts correspond roughly to spatial locations. Similarly, a layer could be interpreted as one quality increment for the entire full resolution



▲ 11. Conceptual correspondence between the spatial and the bit stream representations.

Once the entire image has been compressed, a post-processing operation passes over all the compressed code blocks. This operation determines the extent to which each code block's embedded bit stream should be truncated to achieve a particular target bitrate or distortion. The first, lowest layer (of lowest quality), is formed from the optimally truncated code block bit streams in the manner described above. Each subsequent layer is formed by optimally truncating the code block bit streams to achieve successively higher target bit rates [66]–[68].

The JPEG 2000 standard exhibits many nice features, the most significant being the possibility to define ROI in an

The ROI coding scheme in Part I of the standard is based on the so-called MAXSHIFT method of Christopoulos et al. [14]-[16]. The MAXSHIFT method is an extension of the general ROI scaling-based coding method of [6]. The principle of the general ROI scaling-based method is to scale (shift) coefficients so that the bits associated with the ROI are placed in higher bit planes than the bits associated with the background as depicted in Fig. 12. Then, during the embedded coding process, the most significant ROI bit planes are placed in the bit stream before any background bit planes of the image. Depending on the scaling value, some bits of the

In JPEG 2000, the general scaling-based method is implemented as follows:

- ▲ 1) The wavelet transform is calculated.
- ▲ 2) If an ROI has been defined, then an ROI mask is derived, indicating the set of coefficients that are required for up to lossless ROI reconstruction (Fig. 13).
- ▲ 3) The wavelet coefficients are quantized. Quantized coefficients are stored in a sign magnitude representation. Magnitude bits comprise the most significant part of the implementation precision used (one of the reasons for this is to allow for downscaling of the background coefficients).
- ▲ 4) The coefficients that lay out of the ROI are downscaled by a specified scaling value.



▲ 5) The resulting coefficients are progressively entropy encoded (with the most significant bit planes first).

The decoder reverses these steps to reconstruct the image (Step 2 is still performed before Step 3). As overhead information, the scaling value assigned to the ROI and the coordinates of the ROI are added to the bit stream. The decoder performs also the ROI mask generation but scales up the background coefficients in order to recreate the original coefficients.

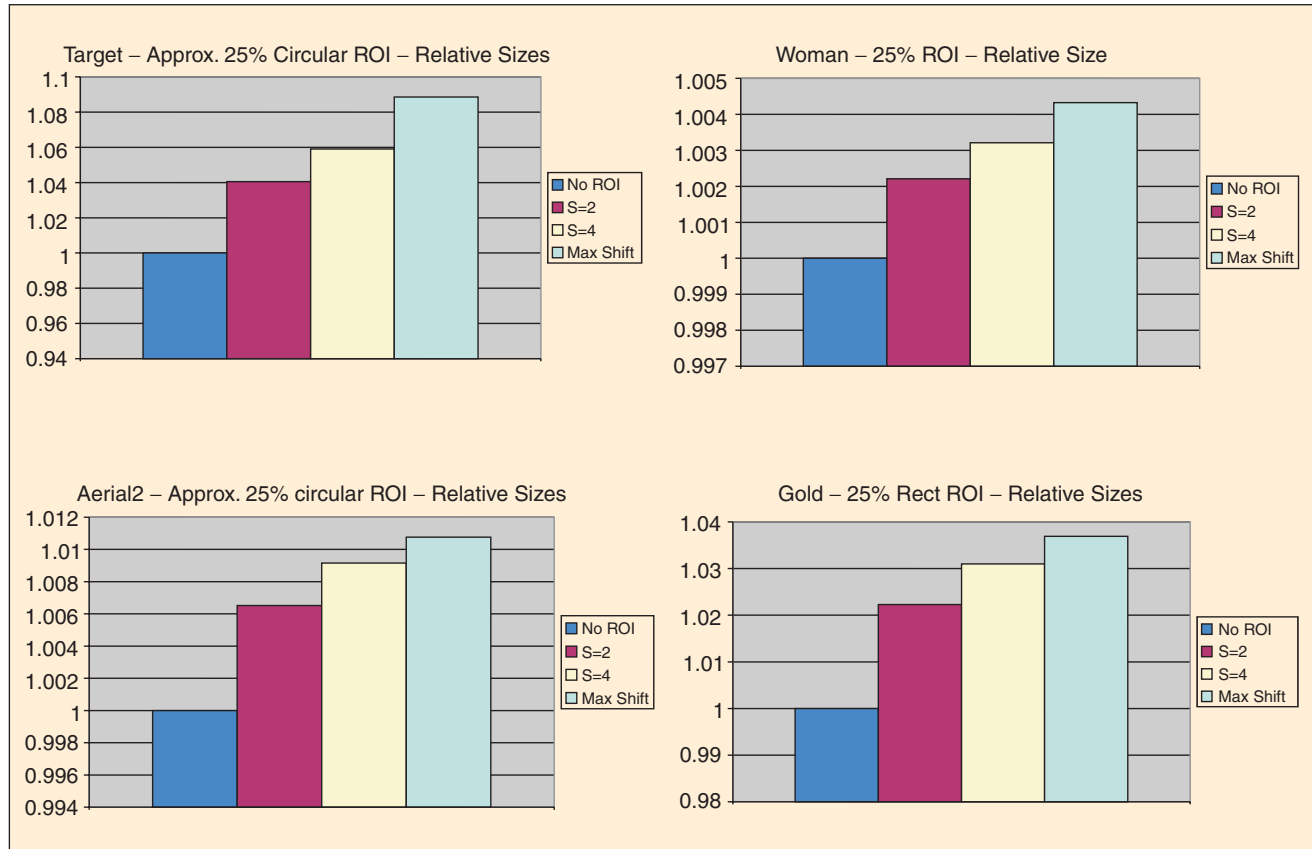
According to the MAXSHIFT method, which is used in Part I of the JPEG 2000 standard, the scaling value is computed in such a way that it makes possible to have arbitrary shaped ROIs without the need for transmitting shape information to the decoder. This means also that the decoder does not have to perform ROI mask generation either (this might still be needed at the encoder). The encoder scans the quantized coefficients and chooses a scaling value S such that the minimum coefficient belonging to the ROI is larger than the maximum coefficient of the background (non-ROI area). The decoder receives the bit stream and starts the decoding process. Every coefficient that is smaller than S belongs to the background and is therefore scaled up. The decoder needs only to up-scale the received background coefficients.

The advantages of the MAXSHIFT method, as compared to the general scaling-based method, is that encoding of arbitrary shaped ROIs is now possible without the need for shape information at the decoder (i.e., no shape

decoder is required) and without the need for calculating the ROI mask. The encoder is also simpler, since no shape encoding is required. The decoder is almost as simple as a nonROI capable decoder, while it can still handle ROIs of arbitrary shape.

In the MAXSHIFT method, since the bit planes with information belonging to the ROI are completely separated from those belonging to the background, the number of bit planes for the ROI and for the background can be chosen independently. This gives the possibility to choose different bit rates for the ROI and for the background. To do this, it is sufficient to discard the least significant bit planes of the ROI and background. With the general scaling-based mode, it is not possible to control these numbers independently.

Experiments have shown that for the lossless coding of images with ROIs, the MAXSHIFT method increases the bit rate by approximately 1% in comparison with the lossless coding of the image without ROI [14]-[16]. This figure is even smaller compared to the general scaling-based method, depending on the scaling value used. This is true for large images (larger than $2K \times 2K$) and for ROI sizes of about 25% of the image. Such an overhead is indeed small, given the fact that the general scaling-based method for arbitrary shaped ROI would require shape information to be transmitted to the decoder, thus increasing the bit rate (in addition to the need of shape encoder/decoder and ROI mask generation at the de-



▲ 14. Lossless coding of images for different scaling factors and different ROI shapes as compared to the non-ROI case. (Image sizes in pixels: aerial 2048×2048 , woman 2048×2560 , target 512×512 , gold 720×576).

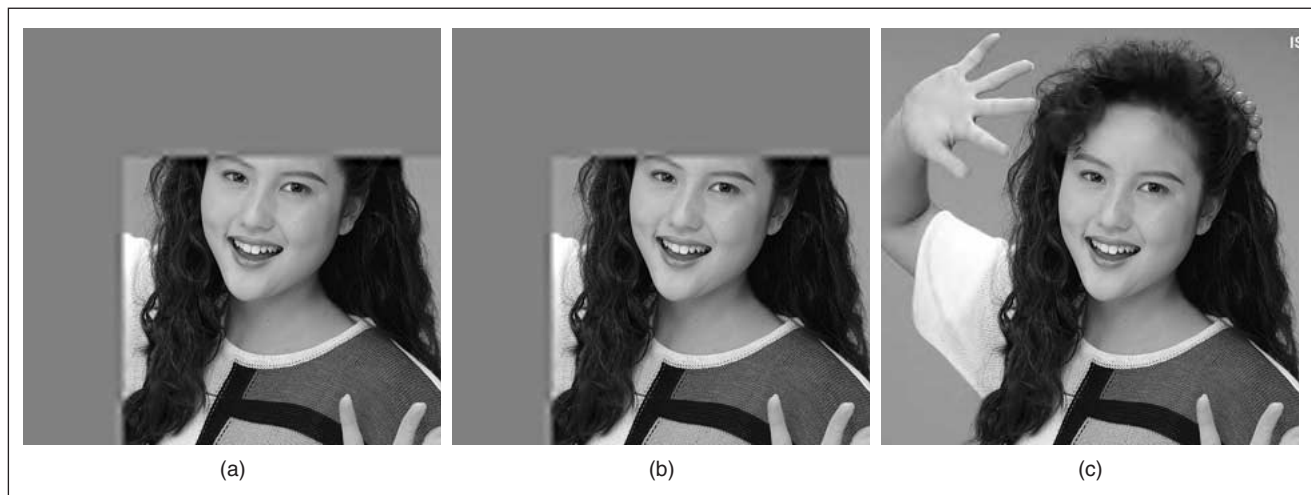
coder side). The performance of the MAXSHIFT method and of the general scaling-based method for different scaling factors, as compared to the lossless coding of an image without ROI, are depicted in Fig. 14. The ROI shape is circular for the aerial and target images and rectangular for the woman and the gold images. It is seen that the MAXSHIFT method results in a very small increase in the bit rate, compared to the general scaling-based method. In fact, for arbitrary shaped regions, where shape information needs to be included in the bit stream, the general scaling-based method and the MAXSHIFT method achieve similar bit rates.

The MAXSHIFT method allows the implementers of an encoder to exploit a number of functionalities that are supported by a compliant decoder. For example, it is possible to use the MAXSHIFT method to encode an image with different quality for the ROI and the background. The image is quantized so that the ROI gets the desired quality (lossy or lossless) and then the MAXSHIFT method is applied. If the image is encoded

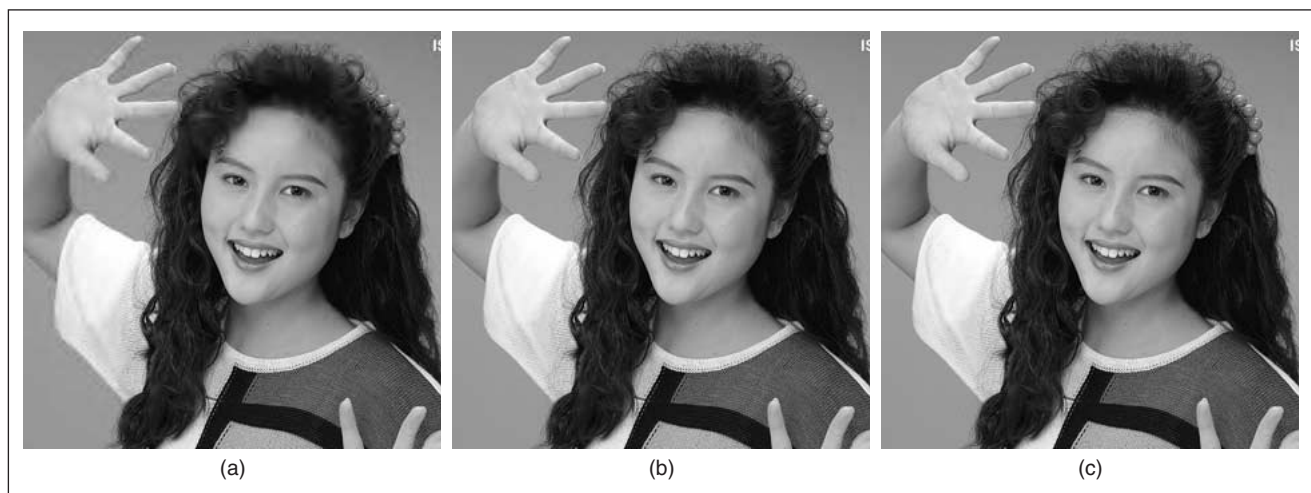
in a progressive by layer manner, not all of the layers of the wavelet coefficients belonging to the background need to be encoded. This corresponds to using different quantization steps for the ROI and the background. Fig. 15 shows an example of ROI coding with the MAXSHIFT method. The ROI is used in all subbands, that is why at the early stages of the transmission, not enough information is used for the background. For comparison purposes, the same result is shown in Fig. 16 for the general scaling-based method, with the scaling value set to six. Similar results can be obtained with the MAXSHIFT method if the few low-resolution subbands are considered as full ROIs. The results show that the MAXSHIFT method can give similar results to the general scaling method, without the need of shape information and mask generation at the decoder.

More about ROI coding

ROI coding is a process performed at the encoder. The encoder decides which is the ROI to be coded in better



▲ 15. ROI encoding results by means of the MAXSHIFT method. Part of the decompressed image “woman” at: (a) 0.5 b/p, (b) 1 b/p, and (c) 2 b/p.

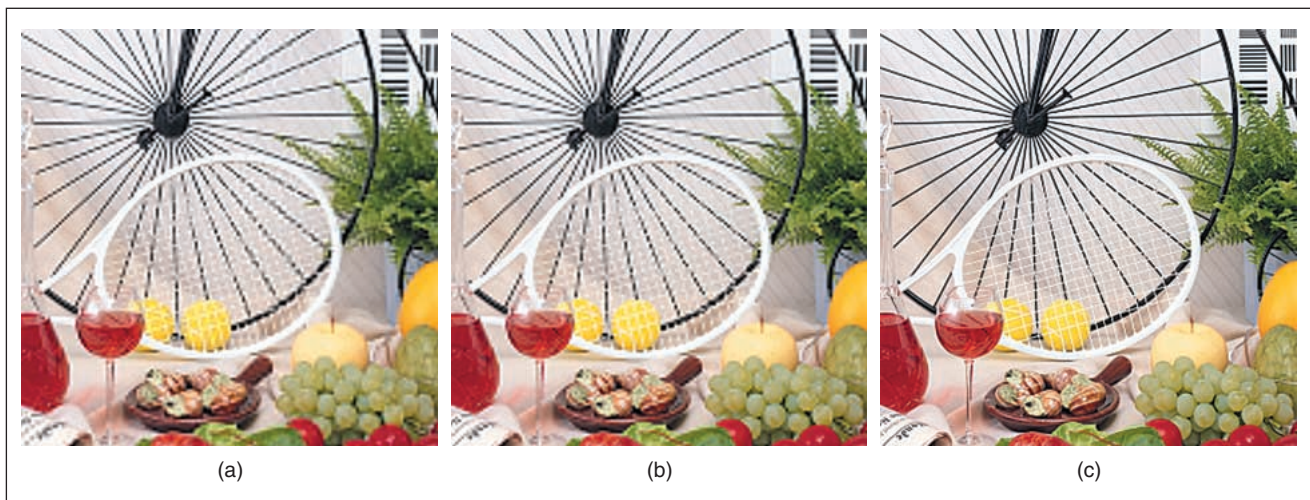


▲ 16. ROI encoding results by means of the general scaling method. Part of the decompressed image “woman” at: (a) 0.5 b/p, (b) 1 b/p, and (c) 2 b/p.

quality than the background. If the ROIs, though, are not known to the encoder in advance, there is still possibility for the decoder to receive only the data that is requested. (A method for interactive ROI selection is described in [60].) Although the simplest method is tiling, this requires that the image be encoded in tiling mode. Another way is to extract packet partitions from the bit stream. This can be done easily, since the length information is stored in the header. Due to the filter impulse response lengths, care has to be taken to extract all data required to decode the ROI. Fine grain access can be achieved by parsing individual code blocks. As in the case of packet partition precincts, it is necessary to determine which code blocks affect which pixel locations (since a single pixel can affect four different code blocks within each subband and each resolution and each component). We

can determine the correct packet affecting these code blocks from the progression order information. The location of the compressed data for the code blocks can be determined by decoding the packet headers. This might be easier than operating the arithmetic coder and context modeling to decode the data [48].

The procedure of coefficient scaling might, in some cases, cause overflow problems due to the finite implementation precision. In JPEG 2000 this problem is minimized since the background coefficients are scaled down, rather than scaling up the ROI coefficients. Thus, if the implementation precision is exceeded only the least significant bit planes of the background are lost (the decoder or the encoder will ignore this part). The advantage is that the ROI, which is considered to be the most important part of the image, is still optimally treated, while the qual-



▲ 17. Example of SNR scalability. Part of the decompressed image "bike" at (a) 0.125 b/p, (b) 0.25 b/p, and (c) 0.5 b/p.



▲ 18. Example of the progressive-by-resolution decoding for the color image "bike."

ity of the background is allowed to be degraded, as it is considered visually less important.

The ROI general scaling-based method can be applied to any embedded coding scheme, as for example, the embedded DCT based coders [52], the various wavelet filters [31] and the zero-tree coders [57], [58], [61].

Scalability

Scalable coding of still images means the ability to achieve coding of more than one qualities and/or resolutions simultaneously. Scalable image coding involves generating a coded representation (bit stream) in a manner which facilitates the derivation of images of more than one quality and/or resolution by scalable decoding. Bit-stream scalability is the property of a bit stream that allows decoding of appropriate subsets of the bit stream to generate complete pictures of quality and/or resolution commensurate with the proportion of the bit stream decoded. Decoders of different complexities (from low performance to high performance) can coexist for a scalable bit stream. While low performance decoders may decode only small portions of the bit stream producing basic quality, high performance decoders may decode much more and produce significantly higher quality. The most important types of scalability are SNR scalability and spatial or resolution scalability. The JPEG 2000 compression system supports scalability. A key advantage of scalable compression is that the target bit rate or reconstruction resolution need not be known at the time of compression. A related advantage of practical significance is that the image need not be compressed multiple times to achieve a target bit rate, as is common with the JPEG compression standard. An additional advantage of scalability is its ability to provide resilience to transmission errors, as the most important data of the lower layer can be sent over the channel with better error performance, while the less critical enhancement layer data can be sent over the channel with poor error performance. Both types of scalability are very important for Internet and database access applications and bandwidth scaling for robust delivery. The SNR and spatial scalability types include the progressive and hierarchical coding modes defined in the JPEG, but they are more general.

SNR Scalability

SNR scalability is intended for use in systems with the primary common feature that a minimum of two layers of image quality is necessary. SNR scalability involves generating at least two image layers of the same spatial resolution, but different qualities, from a single image source. The lower layer is coded by itself to provide the basic image quality and the enhancement layers are coded to enhance the lower layer. An enhancement layer, when added back to the lower layer, regenerates a higher quality reproduction of the input image. Fig. 17 illustrates an example of SNR scalability. The image is first losslessly compressed and decompressed at 0.125 b/p, 0.25 b/p, and 0.5 b/p.

Spatial Scalability

Spatial scalability is intended for use in systems with the primary common feature that a minimum of two layers of spatial resolution is necessary. Spatial scalability involves generating at least two spatial resolution layers from a single source such that the lower layer is coded by itself to provide the basic spatial resolution and the enhancement layer employs the spatially interpolated lower layer and carries the full spatial resolution of the input image source. Fig. 18 shows an example of three levels of progressive-by-resolution decoding for the test image "bike." Spatial scalability is useful for fast database access as well as for delivering different resolutions to terminals with different capabilities in terms of display and bandwidth.

JPEG 2000 supports also a combination of spatial and SNR scalability. It is possible therefore to progress by spatial scalability at a given (resolution) level and then change the progression by SNR at a higher level. This order in progression allows a thumbnail to be displayed first, then a screen resolution image and then an image suitable for the resolution of the printer. It is evident that SNR scalability at each resolution allows the best possible image to be displayed at each resolution.

Notice that the bit stream contains markers that identify the progression type of the bit stream. The data stored in packets are identical regardless of the type of scalability used. Therefore it is trivial to change the progression type or to extract any required data from the bit stream. To change the progression from SNR to progressive by resolution, a parser can read the markers, change the type of progression in the markers, and then write the new markers in the new order. In this manner, fast transcoding of the bit stream can be achieved by a server or gateway, without requiring the use of image decoding and re-encoding, not even the employment of the MQ coder. The required complexity corresponds to that of a copy operation.

In a similar fashion, applications that require the use of a gray scale version of a color compressed image, as for example printing a color image to a gray-scale printer, do not need to receive all color components. A parser can read the markers from the color components and write the markers for one of the components (discarding the packets that contain the color information) [24], [48].

Error Resilience

Error resilience is one of the most desirable properties in mobile and Internet applications [25]. JPEG 2000 uses a variable-length coder (arithmetic coder) to compress the quantized wavelet coefficients. Variable-length coding is known to be prone to channel or transmission errors. A bit error results in loss of synchronization at the entropy decoder and the reconstructed image can be severely damaged. To improve the performance of transmitting compressed images over error prone channels, error resilient b/p and tools are included in the standard. The error resilience tools deal with channel errors using the following approaches: data partitioning and resynchronization, error detection

and concealment, and quality of service (QoS) transmission based on priority [38], [47], [49]. Error resilience is achieved at the entropy coding level and at the packet level. Table 5 summarizes the various ways this is achieved [38].

Entropy coding of the quantized coefficients is performed within code blocks. Since encoding and decoding of the code blocks are independent processes, bit errors in the bit stream of a code block will be restricted within that code block. To increase error resilience, termination of the arithmetic coder is allowed after every coding pass and the contexts may be reset after each coding pass. This allows the arithmetic decoder to continue the decoding process even if an error has occurred.

The lazy coding mode is also useful for error resilience. This relates to the optional arithmetic coding bypass in which bits are fed as raw bits into the bit stream without arithmetic coding. This prevents the error propagation types to which variable length coding is susceptible.

At the packet level, a packet with a resynchronization marker allows spatial partitioning and resynchronization. This is placed in front of every packet in a tile with a sequence number starting at zero and incremented with each packet.

Visual Frequency Weighting

The human visual system plays an important role in the perceived image quality of compressed images [54], [74]. System designers and users should be able to take advantage of the current knowledge in visual perception, i.e., to utilize models of the visual system's varying sensitivity to spatial frequencies, as measured in the contrast sensitivity function (CSF). Since the CSF weight is determined by the visual frequency of the transform coefficient, there will be one CSF weight per subband in the wavelet transform [76], [77]. The design of the CSF weights is an encoder issue and depends on the specific viewing condition under which the decoded image is to be viewed.

Two types of visual frequency weighting are supported by the JPEG 2000 standard. The fixed visual weighting (FVW) and the visual progressive coding or visual progressive weighting (VPW). In the FVW, only one set of CSF weights is chosen and applied in accordance with the viewing conditions. In the VPW, different sets of CSF weights are used at the various stages of the embedded coding. This is because during a progressive transmission stage, the image is viewed at various distances. For example, at low bit rates, the image is viewed from a relatively large distance, while as more and more bits are received and the quality is improved, the viewing distance is decreased (the user is more interested in details and the viewing distance is decreased or the image is magnified, which is equivalent to reducing the viewing distance). FVW can be considered as a special case of VPW.

New File Format with IPR Capabilities

The JPEG 2000 (JP2) file format commences with a unique signature and contains the size of the image, the

bit depth of the components in the file in cases where the bit depth is not constant across all components, the color space of the image, the palette which maps a single component in index space to a multiple-component image, the type and ordering of the components within the code stream, the resolution of the image, the resolution at which the image was captured, the default resolution at which the image should be displayed, the code stream, intellectual property information about the image, a tool by which vendors can add XML formatted information to a JP2 file, and other information [8], [38], [39].

The JP2 file format is optional in the standard. The JP2 format provides a foundation for storing application specific data (metadata) in association with a JPEG 2000 code stream, such as information required to display the image. This format has got provisions for both image and metadata and specifies mechanisms to indicate image properties, such as the tone-scale or color-space of the image, to recognize the existence of intellectual property rights (IPR) information in the file and to include metadata (as for example vendor specific information). Metadata give the opportunity to the reader to extract information about the image, without having to decode it, thus allowing fast text based search in a database.

In addition to specifying the color space, the standard allows for the decoding of single component images, where the value of that single component represents an index into a palette of colors. An input of a decompressed sample to the palette converts the single value to a multiple-component tuple. The value of that tuple represents the color of the sample.

Performance Comparisons

To judge the efficiency of the JPEG 2000 as compared to other standards, extensive comparisons have been conducted with regard to lossy performance, lossless performance, resilience to errors, and other characteristics. The algorithms have been evaluated on several images, mainly from the JPEG 2000 test set, which cover a wide range of imagery types.

The Software

The software used for the experiments is reported in Table 6. The Verification Model 8.6 was used for the JPEG

Table 5. Tools for Error Resilience.

Type of Tool	Name
Entropy coding level	<ul style="list-style-type: none"> —code blocks —termination of the arithmetic coder for each pass —reset of contexts after each coding pass —selective arithmetic coding bypass —segmentation symbols
Packet level	<ul style="list-style-type: none"> —short packet format —packet with resynchronization marker

Table 6. Software Implementations.

Standard	Name of the Software	Source Code	Software Developers	Available at
JPEG 2000	Verification Model 8.6	C		http://www.jpeg.org (*)
JPEG 2000	JasPer	C	Image Power Univ. of British Columbia	http://www.ece.ubc.ca/mdadams/jasper http://spmng.ece.ubc.ca http://www.imagepower.com
JPEG 2000	JJ2000	JAVA	Cannon Research EPFL Ericsson	http://jj2000.epfl.ch
JPEG		C	Independent JPEG Group	http://www.ijg.org
JPEG-LS	SPMG	C	Univ. of British Columbia	http://spmng.ece.ubc.ca
Lossless JPEG		C	Cornell University	ftp://ftp.cs.cornell.edu/pub/multimed
PNG	PNG	C		ftp://ftp.uu.net/graphics.png
(*) Available to members only				

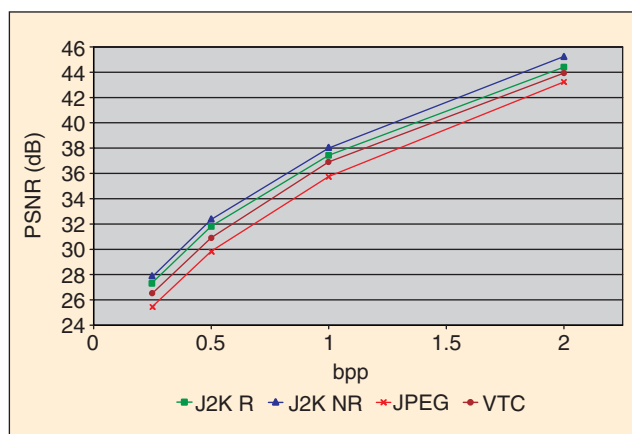
2000 compression and decompression [39]. The other implementation software was the Independent JPEG Group (version 6b), the SPMG JPEG-LS (version 2.2), the lossless JPEG (version 1.0) [30], the libpng of PNG (version 1.03) [72], and the MPEG-4 MoMuSys Verification Model of August 1999 [42].

Lossy Compression Results

The objective (PSNR) comparison results between the different standards are graphically shown in Fig. 19. It is seen that the lossy (nonreversible) JPEG 2000 (J2K_{NR}) outperforms all other standards. Compared to the JPEG standard, it is realized that JPEG 2000 performs better by approximately 2 dB for all compression ratios. As expected, the lossless (reversible) JPEG 2000 (J2K_R) does

not perform as well, due to the use of reversible wavelet filters, i.e., the 5-tap/3-tap filters.

The superiority of the JPEG 2000 over the existing JPEG can be subjectively judged with the help of Fig. 20 and Fig. 21, where the reconstructed grayscale image “watch” and the color image “ski” are shown after compression at 0.2 b/p and 0.25 b/p, respectively. Results of the image cmpnd1 compressed by means of JPEG baseline and JPEG 2000 at 0.5 b/p, are shown in Fig. 22. It becomes evident that JPEG 2000 outperforms JPEG baseline. This performance superiority of the JPEG 2000 decreases as the bit rate increases. In fact, from the compression point of view, JPEG 2000 gives about 10 to 20% better compression factors compared to JPEG baseline, for a bit rate of about 1 b/p. Visual comparisons of JPEG compressed images (baseline JPEG with optimized Huffman tables) and JPEG 2000 compressed images showed that for a large category of images, JPEG 2000 file sizes were on average 11% smaller than JPEG at 1.0 b/p, 18% smaller at 0.75 b/p, 36% smaller at 0.5 b/p and 53% smaller at 0.25 b/p [37]. In general, we can say that for high quality imaging applications (i.e., 0.5-1.0 b/p) JPEG 2000 is 10-20% better than JPEG.



▲ 19. PSNR results for the lossy compression of a natural image by means of different compression standards.

Lossless Compression Results

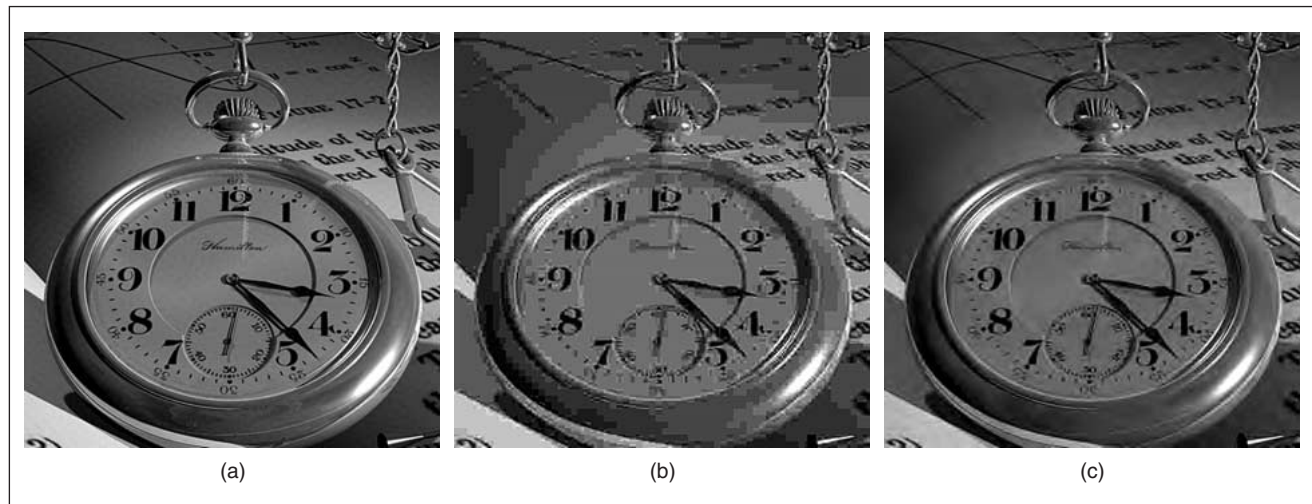
The lossless compression efficiency of the reversible JPEG 2000 (J2K_R), JPEG-LS, lossless JPEG (L-JPEG), and PNG is reported in Table 7 [59]. It is seen that JPEG 2000 performs equivalently to JPEG-LS in the case of the natural images, with the added benefit of scalability. JPEG-LS, however, is advantageous in the case of the compound image. Taking into account that JPEG-LS is

significantly less complex than JPEG 2000, it is reasonable to use JPEG-LS for lossless compression. In such a case though, the generality of JPEG 2000 is sacrificed.

Error Resilience Results

The evaluation results of JPEG 2000 in error resilience are shown in Table 8 [19], [59]. A transmission channel with random errors has been simulated, and the average reconstructed image quality after decompression has been measured. In the case of JPEG, the results were 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 information was moved to the bit-stream header (using the PPM marker [38]) and the entropy coded data had been protected by the regular termination of the arithmetic coder combined with the error resilience termination and segment symbols (the overhead for these protections is less than 1%).

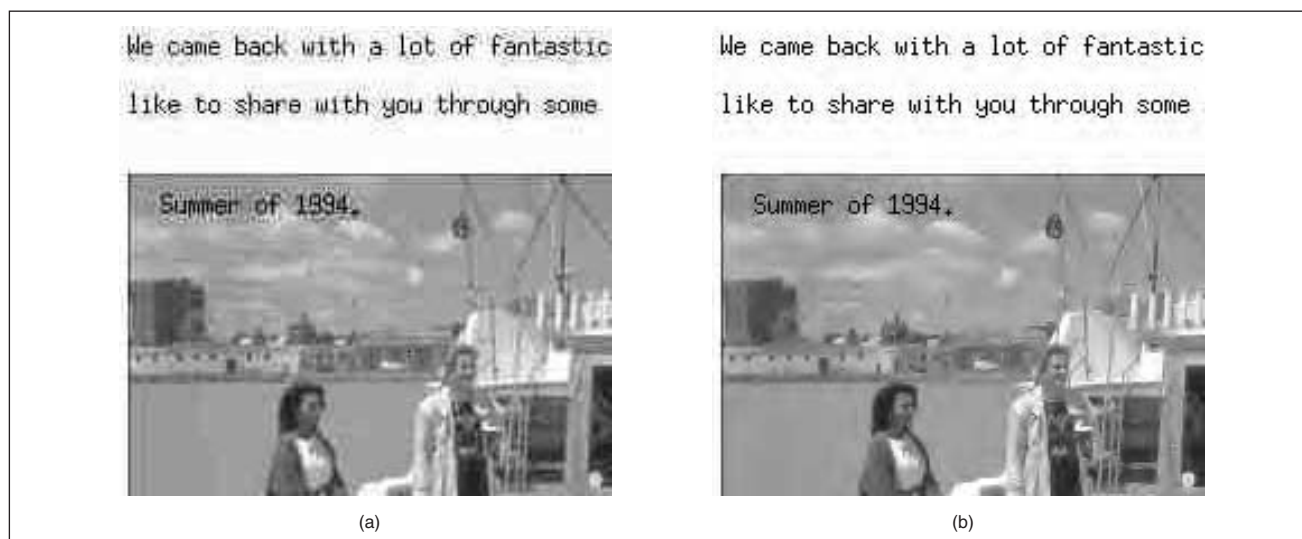
In both cases the bit-stream header was transmitted without errors. As can be deduced from Table 8, the reconstructed image quality under transmission errors is higher for JPEG 2000 compared to that of JPEG. However, at low bit rates (0.25 and 0.5 b/p), the quality of JPEG 2000 decreases more rapidly than JPEG as the error rate increases. It is interesting to observe that at higher error rates (i.e., $1e-4$), the reconstructed image quality in JPEG 2000 is almost constant across all bit rates. This is due to the fact that in JPEG 2000 each subband block is coded by bit planes. When the error rate is high enough almost all blocks are affected in the most significant bit planes, which are transmitted first. When a particular bit plane is affected in a block, lower bit planes cannot be decoded and are therefore of no use. In the case of JPEG the problem is even worse: the higher the encoding bit-rate the lower the decoded quality. This can be explained by the fact that when an 8×8 block is affected by a transmission error the entire block is basically lost. The higher the en-



▲ 20. Image “watch” of size 512×512 (courtesy of Kevin Odhner): (a) original, and reconstructed after compression at 0.2 b/p by means of (b) JPEG and (c) JPEG 2000.



▲ 21. Reconstructed image “ski” after compression at 0.25 b/p by means of (a) JPEG and (b) JPEG 2000.



▲ 22. Part of the reconstructed image "cmpnd1" after compression at 0.5 b/p by means of (a) JPEG and (b) JPEG 2000.

coding bit rate, the more the bits needed to code a block, and therefore the probability of a block being hit by an error and lost is higher for the same bit error rate. In other words, in JPEG the density of error protection decreases with an increase in bit rate.

Features and Functionalities

Table 9 summarizes the comparative results of JPEG, JPEG-LS, and JPEG 2000 from the functionality point of view. The number of the bullet symbols indicates the strength by which each functionality is supported. The Verification Model 8.6 of JPEG 2000 has been used in this case.

MPEG-4 VTC and JPEG 2000 produce progressive bit streams. JPEG 2000 provides bit streams that are parseable and can easily be reorganized by a transcoder on the fly. JPEG 2000 also allows random access (with minimal decoding) to the block-level of each subband, thus making possible to decode a region of an image without

decoding the whole image. Notice that MPEG-4 supports coding of arbitrary shape objects by means of an adaptive DWT, but it does not support lossless coding [27], [75].

Notice that DCT based algorithms can also achieve many of the features of JPEG 2000, as ROI, embedded bit stream, etc. [6], [14]-[16], [31], [51], [52], [53], [60]. DCT based coding schemes, however, due to the block-based coding nature, cannot perform well at low rates, unless post-processing operations are involved [7], [32]. The complexity of those schemes is increased compared to baseline JPEG and their compression performance is not better than wavelet based coding schemes (although very close). Additionally, although JPEG 2000 offers better performance than JPEG, different types of artifacts appear in wavelet based coders. Some results on post-processing of JPEG 2000 compressed images for tiling and ringing artifact reduction, have already been reported [34], [36].

Overall, the JPEG 2000 standard offers the richest set of features in a very efficient way and within a unified algorithm [19], [59], [63]. However, this comes at a price of additional complexity in comparison to JPEG and JPEG-LS. This might be perceived as a disadvantage for some applications, as was the case with JPEG when it was first introduced.

A Note on the Future Parts of the Standard

Part I of the JPEG 2000 standard was studied here. Part I describes the core coding system, which should be used to provide maximum interchange. Part II consists of optional technologies not required for all implementations. Evidently, images encoded with Part II technology will not be able to be decoded with Part I decoders. As an example, Part II will include trellis coded quantization [38], [44], user defined wavelets, wavelet packets and other decompositions, general scaling-based ROI coding

Table 7. Lossless Compression Ratios.				
	J2K _R	JPEG-LS	L-JPEG	PNG
aerial2	1.47	1.51	1.43	1.48
bike	1.77	1.84	1.61	1.66
cafe	1.49	1.57	1.36	1.44
chart	2.60	2.82	2.00	2.41
cmpnd1	3.77	6.44	3.23	6.02
target	3.76	3.66	2.59	8.70
us	2.63	3.04	2.41	2.94
average	2.50	2.98	2.09	3.52

method, etc. Part III will define motion JPEG 2000 (MJP2) and will be based on Part I of JPEG 2000 [21]. MJP2 will be used in many different areas, as for example in applications where it is desired to have a single codec for both still pictures and motion sequences (which is a common feature of digital still cameras), or in applications where very high quality motion pictures are required (e.g., medical imaging and motion picture production), or in video applications in error prone environments (e.g., wireless and the Internet). The standard will allow one or more JPEG 2000 compressed image sequences, synchronized audio and metadata to be stored in the Motion JPEG 2000 file format (MJ2). Finally, Motion JPEG 2000 is targeting interoperability with the JPEG 2000 file format (JP2) and the MPEG-4 file format (MP4). Part IV of the standard will define the conformance testing. Part V will define the reference software as high quality free software. Currently, two reference software implementations do exist, the JJ2000 software in Java and the JasPer software in C (see Table 6) [1]. Part VI will define a compound image file format. The work plan of these parts is shown in Table 10 [24], [40], [41].

As of this writing several new parts are under consideration by the JPEG 2000 committee to enhance its features. These are Part VIII for a reference hardware implementation, Part IX for mobile wireless specific application features (JP3G), Part X for secure JPEG 2000 bit streams (JPSEC), Part XI for interactivity tools: APIs and protocols (JPIP), and Part XII for volumetric data (JP3D). In addition to the above, various amendments to different parts of JPEG 2000 standard are in preparation. These currently include two amendments to part I, one amendment to part II, and one amendment to part V.

Conclusions

JPEG 2000 is the new standard for still image compression. It provides a new framework and an integrated toolbox to better address increasing needs for compression. It also provides a wide range of

functionalities for still image applications, like Internet, color facsimile, printing, scanning, digital photography, remote sensing, mobile applications, medical imagery, digital library, and E-commerce. Lossless and lossy coding, embedded lossy to lossless, progressive by resolution and quality, high compression efficiency, error resilience, and lossless color transformations are some of its characteristics. Comparative results have shown that JPEG 2000 is indeed superior to established still image com-

Table 8. PSNR results in dB for 200 runs of the decoded cafe image transmitted over a noisy channel for various bit error rates (ber) and compression rates for the JPEG baseline and the JPEG 2000.

b/p	ber	0	1e-6	1e-5	1e-4
0.25	JPEG 2000	23.06	23.00	21.62	16.59
	JPEG	21.94	21.79	20.77	16.43
0.5	JPEG 2000	26.71	26.42	23.96	17.09
	JPEG	25.40	25.12	22.95	15.73
1.0	JPEG 2000	31.90	30.75	27.08	16.92
	JPEG	30.84	29.24	23.65	14.80
2.0	JPEG 2000	38.91	36.38	27.23	17.33
	JPEG	37.22	30.68	20.78	12.09

Table 9. Functionality Evaluation Results.

	JPEG 2000	JPEG-LS	JPEG	MPEG-4 VTC
Lossless compression performance	•••	••••	•	
Lossy compression performance	•••••	•	•••	••••
Progressive bitstreams	••••		•	••
Region of Interest (ROI) coding	•••			•
Arbitrary shaped objects				••
Random access	••			
Low complexity	••	•••••	•••••	•
Error resilience	•••	•	•	•••
noniterative rate control	•••			•
Genericity	•••	•••	••	••
A bullet indicates that the corresponding functionality is supported. The number of bullet symbols characterizes the degree of support.				

Table 10. Schedule of JPEG 2000 Parts.							
Part	Title	Call for Papers	Working Draft	Committee Draft	Final Committee Draft	Final Draft International Standard	International Standard
I	JPEG 2000 Image Coding System: Core Coding System	97/03	99/03	99/12	00/03	00/10	00/12
II	JPEG 2000 Image Coding System: Extensions	97/03	00/03	00/08	00/12	01/07	01/10
III	Motion JPEG 2000	99/12	00/07	00/12	01/03	01/07	01/10
IV	Conformance Testing	99/12	00/07	00/12	01/03	01/11	02/03
V	Reference Software	99/12	00/03	00/07	00/12	01/08	01/11
VI	Compound Image File Format	97/03	00/12	01/03	01/11	02/03	02/05

pression standards. Work is still needed for optimizing its implementation performance. At the time of writing several hardware and software products compliant with the JPEG 2000 Part I standard have been announced. A list of them can be obtained from <http://jpeg2000.epfl.ch> as well as other relevant and useful information.

Acknowledgments

The authors would like to thank J. Kovačević and the anonymous reviewers for their constructive comments. Various inputs from Diego Santa Cruz and Rafael Grosbois for the performance comparison section are acknowledged.

Athanasios Skodras received the B.Sc. degree in physics from Aristotle University of Thessaloniki, Greece, in 1980, the Diploma degree in Computer Engineering and Informatics in 1985, and the Ph.D. degree in electronics in 1986 from the University of Patras, Greece. Since 1986 he has been with the Computer Technology Institute, Patras, Greece, and the University of Patras, where currently he is an Associate Professor at the Electronics Laboratory. In 1988-1989 and 1996-1997 he was a Visiting Researcher at Imperial College, London, UK. His current research interests include image and video coding, digital watermarking, fast transform algorithms, real-time digital signal processing, and multimedia applications. He has authored or co-authored over 60 technical papers and four books and holds two international patents. He is an Associate Editor for the *IEEE Transactions on Circuits and Systems II* and *Pattern Recognition*. He is the Chair of the IEEE CAS & SSC Chapters in Greece and the Technical Coordinator of the WG6 on image and video coding of the Hellenic Organization for Standardization. He was the Technical Program Chair for the 13th

IEEE International Conference on Digital Signal Processing (DSP-97). He is the co-reipient of the 2000 Chester Sall Award in the *IEEE Transactions on Consumer Electronics*. He is a Chartered Engineer, Senior Member of the IEEE, and member of the IEE, EURASIP, and the Technical Chamber of Greece

Charilaos Christopoulos obtained his B.Sc. in physics from the University of Patras in 1989 his M.Sc. in software engineering from the University of Liverpool, UK, in 1991 and his Ph.D. in video coding from the University of Patras in 1996. From 1993 to 1995 he was a research fellow at the Free University of Brussels. He joined Ericsson Research in 1995 where he is now Manager of Ericsson's MediaLab. He has been head of Swedish delegation in ISO/SC29/WG01 (JPEG/JBIG), editor of the Verification Model of JPEG 2000, and co-editor of the JPEG 2000 standard in ISO. He holds 15 Swedish filed/granted patents. He received the 2000 Chester W. Sall Award in the *IEEE Transactions on Consumer Electronics* and third place for the same award in 1998. His research interests include image and video processing, security and watermarking technologies, mobile communications, 3-D and virtual/augmented reality. He is a Senior Member of the IEEE, Associate Editor of *IEEE Transactions on Circuits and Systems II*, member of the editorial board of *Signal Processing*, and member of the European Association for Speech, Signal and Image Processing (EURASIP) and the Hellenic Society of Physicists.

Touradj Ebrahimi received his M.Sc. and Ph.D. degrees in electrical engineering from the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, in 1989 and 1992, respectively. In 1993, he was with Sony Corporation in Tokyo. In 1994, he served as a research con-

sultant at AT&T Bell Laboratories. He is currently a Professor at EPFL. He has been the recipient of various distinctions such as the IEEE and Swiss national ASE award, the SNF-PROFILE grant for advanced researchers, two ISO-Certificates for key contributions to MPEG-4 and JPEG 2000, and the best paper award of *IEEE Transactions on Consumer Electronics*. He is head of Swiss delegation to MPEG JPEG and SC29. He is Associate Editor of *IEEE Transactions on Image Processing*, *SPIE Optical Engineering Magazine*, and *EURASIP Image Communication Journal*. His research interests include still, moving, and three-dimensional image processing and coding, and visual information security. He is the author or the co-author of more than 100 research publications and holds ten patents. He is a member of the IEEE, SPIE, and IS&T.

References

- [1] M.D. Adams and F. Kossentini, "JasPer: A software-based JPEG-2000 Codec implementation," in *Proc. IEEE Int. Conf. Image Processing*, Vancouver, Canada, Sept. 2000, vol. II, pp. 53-56.
- [2] M.D. Adams and F. Kossentini, "Reversible integer-to-integer wavelet transforms for image compression: Performance evaluation and analysis," *IEEE Trans. Image Processing*, vol. 9, pp. 1010-1024, June 2000.
- [3] M.D. Adams, "Reversible wavelet transforms and their application to embedded image compression," M.S. thesis, Univ. Victoria, Canada, 1998. Available <http://www.ece.ubc.ca/mdadams/>.
- [4] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using the wavelet transform," *IEEE Trans. Image Processing*, pp. 205-220, Apr. 1992.
- [5] J. Askelof, C. Christopoulos, M. Larsson Carlander, and F. Ojier, "Wireless image applications and next-generation imaging," *Ericsson Review*, no. 2, pp. 54-61, 2001. Available http://www.ericsson.com/review/2001_02/.
- [6] E. Atsumi and N. Farvardin, "Lossy/lossless region-of-interest image coding based on set partitioning in hierarchical trees," *Proc. IEEE Int. Conf. Image Processing*, Chicago, IL, Oct. 1998, pp. 87-91.
- [7] V. Bhaskaran and K. Konstantinides, *Image and Video Compression Standards: Algorithms and Applications*, 2nd ed. Norwell, MA: Kluwer, 1997.
- [8] M. Bolick, J. Scott Houchin, and G. Wu, "JPEG 2000 next generation image compression system features and syntax," in *Proc. IEEE Int. Conf. Image Processing*, Vancouver, Canada, Sept. 2000, vol. II, pp. 45-48.
- [9] A. Bovik, Ed., *Handbook of Image & Video Processing*. San Diego, CA: Academic, 2000.
- [10] C.M. Brislawn, "Classification of nonexpansive symmetric extension transforms for multirate filter banks," *Appl. Computational Harmonic Anal.*, vol. 3, pp. 337-357, 1996.
- [11] C.M. Brislawn, "Preservation of subband symmetry in multirate signal coding," *IEEE Trans. Signal Processing*, vol. 43, pp. 3046-3050, Dec. 1995.
- [12] A.R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo, "Lossless image compression using integer to integer wavelet transforms," in *Proc. IEEE Int. Conf. Image Processing*, vol. 1. Santa Barbara, CA, Oct. 1997, pp. 596-599.
- [13] C.A. Christopoulos, A.N. Skodras, and T. Ebrahimi, "The JPEG 2000 still image coding system: An overview," *IEEE Trans. Consumer Electron.*, vol. 46, pp. 1103-1127, Nov. 2000.
- [14] C.A. Christopoulos, J. Askelof, and M. Larsson, "Efficient methods for encoding regions of interest in the upcoming JPEG 2000 still image coding standard," *IEEE Signal Processing Lett.*, vol. 7, pp. 247-249, Sept. 2000.
- [15] C.A. Christopoulos, J. Askelof, and M. Larsson, "Efficient encoding and reconstruction of regions of interest in JPEG 2000," in *Proc. X European Signal Processing Conf. (EUSIPCO-2000)*, Tampere, Finland, Sept. 2000, pp. 1133-1136.
- [16] C.A. Christopoulos, J. Askelof, and M. Larsson, "Efficient region of interest encoding techniques in the upcoming JPEG 2000 still image coding standard," in *Proc. IEEE Int. Conf. Image Processing (ICIP 2000)*, vol. II. Vancouver, Canada, Sept. 2000, pp. 41-44.
- [17] C. Chrysafis and A. Ortega, "An algorithm for low memory wavelet image compression," in *Proc. IEEE Int. Conf. Image Processing*, vol. III. Kobe, Japan, Oct. 1999, pp. 354-358.
- [18] C. Chrysafis and A. Ortega, "Line-based, reduced memory, wavelet image compression," *IEEE Trans. Image Processing*, vol. 9, pp. 378-389, Mar. 2000.
- [19] T. Ebrahimi, D. Santa Cruz, J. Askelof, M. Larsson, and C. Christopoulos, "JPEG 2000 still image coding versus other standards," in *Proc. SPIE Int. Symp.*, San Diego CA, 30 July - 4 Aug. 2000, pp. 446-454.
- [20] F. Frey and S. Suesstrunk, "Digital photography—How long will it last?," in *IEEE Int. Symp. Circuits and Systems (ISCAS 2000)*, vol. V. Geneva, Switzerland, 28-31 May 2000, pp. 113-116.
- [21] T. Fukuhara, K. Katoh, K. Hosaka, and A. Leung, "Motion-JPEG 2000 standardization and target market," in *Proc. IEEE Int. Conf. Image Processing (ICIP 2000)*, Vancouver, Canada, 10-13 Sept. 2000, vol. II, pp. 57-66.
- [22] M. Ghanbari, *Video Coding: An Introduction to Standard Coders*. London, UK: IEE, 1999.
- [23] J.D. Gibson, T. Berger, T. Lookabaugh, D. Lindbergh, and R.L. Baner, *Digital Compression for Multimedia: Principles & Standards*. San Mateo, CA: Morgan Kaufmann, 1998.
- [24] M.J. Gormish, D. Lee, and M.W. Marcellin, "JPEG 2000: Overview, architecture and applications," in *Proc. IEEE Int. Conf. Image Processing (ICIP 2000)*, Vancouver, Canada, 10-13 Sept. 2000, vol. II, pp. 29-32.
- [25] V.K. Goyal, "Multiple description coding: Compression meets the network," *IEEE Signal Processing Mag.*, vol. 18, pp. 74-93, Sept. 2001.
- [26] V.K. Goyal, "Theoretical foundations of transform coding," *IEEE Sig. Proc. Magazine*, vol. 18, pp. 9-21, Sept. 2001.
- [27] *Information Technology—Coding of Audio-Visual Objects—Part 2: Visual*, ISO/IEC 14496-2:1999, Dec. 1999.
- [28] *JPEG 2000 Part 1: Final Draft International Standard (ISO/IEC FDIS15444-1)*, ISO/IEC JTC1/SC29/WG1 N1855, Aug. 2000.
- [29] *JPEG 2000 Requirements and Profiles*, ISO/IEC JTC1/SC29/WG1 N1271, Mar. 1999.
- [30] *Report on CEV2: Postprocessing for Ringing Artifact Reduction*, ISO/IEC JTC1/SC29/WG1 N1354, July 1999.
- [31] *Information Technology—Coded Representation of Picture and Audio Information—Lossy/Lossless Coding of Bi-level Images*, ISO/IEC JTC1/SC29/WG1 N1359, 14492 Final Committee Draft, July 1999.
- [32] *Post-Processing Approach to Tile Boundary Removal*, ISO/IEC JTC1/SC29/WG1 N1487, Dec. 1999.
- [33] *Visual Evaluation of JPEG 2000 Color Image Compression Performance*, ISO/IEC JTC1/SC29/WG1 N1583, Mar. 2000.
- [34] *JPEG 2000 Verification Model 8.6*, ISO/IEC JTC1/SC29/WG1 N1894, 2000.
- [35] *New Work Item: JPEG 2000 Image Coding System*, ISO/IEC JTC1/SC29/WG1 N390R, Mar. 1997.
- [36] *Call for Contributions for JPEG 2000 (JTC 1.29.14, 15444): Image Coding System*, ISO/IEC JTC1/SC29/WG1 N505, Mar. 1997.
- [37] *JPEG-LS (14495) Final CD*, ISO/IEC JTC1/SC29/WG1 N575, July 1997.
- [38] *Progressive Lossy To Lossless Core Experiment with a Region of Interest: Results with the S, S+P, Two-Ten Integer Wavelets and with the Difference Coding Method*, ISO/IEC JTC1/SC29/WG1 N741, Mar. 1998.

- [39] *Core Experiment on Improving the Performance of the DCT: Results with the Visual Quantization Method, Deblocking Filter and Pre/Post Processing*, ISO/IEC JTC1/SC29/WG1 N742, Mar. 1998.
- [40] *Resolutions of 22nd WG1 New Orleans Meeting*, ISO/IEC JTC1/SC29/WG1 N1959, 8 Dec. 2000.
- [41] *Press Release of the 23rd WG1 Singapore Meeting*, ISO/IEC JTC1/SC29/WG1 N2058, 9 Mar. 2001.
- [42] *MoMuSys VM*, ISO/IEC JTC1/SC29/WG11 N2805, Aug. 1999.
- [43] K. Parulski and M. Rabbani, "The continuing evolution of digital cameras and digital photography systems," in *IEEE Int. Symp. Circuits Syst. (ISCAS 2000)*, vol. V. Geneva, Switzerland, 28-31 May 2000, pp. 101-104.
- [44] J.H. Kasner, M.W. Marcellin, and B.R. Hunt, "Universal trellis coded quantization," *IEEE Trans. Image Processing*, vol. 8, pp. 1677-1687, Dec. 1999.
- [45] J. Kovačević and W. Sweldens, "Wavelet families of increasing order in arbitrary dimensions," *IEEE Trans. Image Processing*, vol. 9, pp. 480-496, Mar. 2000.
- [46] L. Gall and A. Tabatabai, "Subband coding of digital images using symmetric short kernel filters and arithmetic coding techniques," in *Proc. IEEE Int. Conf. ASSP*, NY, 1988, pp. 761-765.
- [47] J. Liang and R. Talluri, "Tools for robust image and video coding in JPEG 2000 and MPEG-4 Standards," in *Proc. SPIE Visual Communications and Image Processing Conf. (VCIP)*, San Jose, CA, Jan. 1999.
- [48] M.W. Marcellin, M. Gormish, A. Bilgin, and M. Bolick, "An overview of JPEG 2000," in *Proc. IEEE Data Compression Conf.*, Snowbird, UT, Mar. 2000, pp. 523-541.
- [49] I. Moccagata, S. Sodagar, J. Liang, and H. Chen, "Error resilient coding in JPEG-2000 and MPEG-4," *IEEE J. Select. Areas Commun. (JSAC)*, vol. 18, pp. 899-914, June 2000.
- [50] M.J. Nadenau and J. Reichel, "Opponent color, human vision and wavelets for image compression," in *Proc. 7th Color Imaging Conf.*, Scottsdale, AZ, 16-19 Nov. 1999, pp. 237-242.
- [51] D. Nister and C. Christopoulos, "Lossless region of interest with a naturally progressive still image coding algorithm," in *Proc. IEEE Int. Conf. Image Processing*, Chicago, IL, Oct. 1998, pp. 856-860.
- [52] D. Nister and C. Christopoulos, "An embedded DCT-based still image coding algorithm," *IEEE Signal Processing Lett.*, vol. 5, pp. 135-137, June 1998.
- [53] D. Nister and C. Christopoulos, "Lossless region of interest with embedded wavelet image coding," *Signal Processing*, vol. 78, no. 1, pp. 1-17, 1999.
- [54] T. O'Rourke and R. Stevenson, "Human visual system based wavelet decomposition for image compression," *J. VCIP* V. 6, pp. 109-121, 1995.
- [55] W.B. Pennebaker and J. L. Mitchell, *JPEG: Still Image Data Compression Standard*. New York: Van Nostrand Reinhold, 1993.
- [56] K.R. Rao and J.J. Hwang, *Techniques and Standards for Image, Video and Audio Coding*. Englewood Cliffs, NJ: Prentice-Hall, 1996.
- [57] A. Said and W.A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 243-250, June 1996.
- [58] A. Said and W.A. Pearlman, "An image multiresolution representation for lossless and lossy compression," *IEEE Trans. Image Processing*, vol. 5, pp. 1303-1310, Sept. 1996.
- [59] D. Santa Cruz and T. Ebrahimi, "An analytical study of the JPEG 2000 functionalities," in *Proc. IEEE Int. Conf. Image Processing (ICIP 2000)*, Vancouver, Canada, 10-13 Sept. 2000, vol. II, pp. 49-52.
- [60] D. Santa Cruz, M. Larsson, J. Askelof, T. Ebrahimi, and C. Christopoulos, "Region of interest coding in JPEG 2000 for interactive client/server applications," in *Proc. IEEE Int. Workshop Multimedia Signal Processing*, Copenhagen, Denmark, Sept. 1999, p. 389-384.
- [61] J.M. Shapiro, "Embedded imaging coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, pp. 3445-3462, Dec. 1993.
- [62] A. Bilgin, P.J. Sementilli, F. Sheng, and M.W. Marcellin, "Scalable image coding using reversible integer wavelet transforms," *IEEE Trans. Image Processing*, vol. 9, pp. 1972-1977, Nov. 2000.
- [63] A.N. Skodras, C. Christopoulos, and T. Ebrahimi, "JPEG 2000: The upcoming still image compression standard," *Pattern Recognition Lett.*, vol. 22, pp. 1337-1345, Oct. 2001.
- [64] W. Sweldens, "The lifting scheme, A custom-design construction of biorthogonal wavelets," *Appl. Comput. Harmonic Anal.*, vol. 3, no. 2, pp. 186-200, 1996.
- [65] W. Sweldens, "The lifting scheme: Construction of second generation wavelets," *SIAM J. Math. Anal.*, vol. 29, no. 2, pp. 511-546, 1997.
- [66] D. Taubman and A. Zalkor, "Multirate 3-D subband coding of video," *IEEE Trans. Image Processing*, vol. 3, pp. 572-578, Sept. 1994.
- [67] D. Taubman, "High performance scalable image compression with EBCOT," in *Proc. IEEE Int. Conf. Image Processing*, vol. III. Kobe, Japan, Oct. 1999, pp. 344-348.
- [68] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Trans. Image Processing*, vol. 9, pp. 1158-1170, July 2000.
- [69] D. Taubman, E. Ordentlich, M. Weinberger, G. Seroussi, I. Ueno, and F. Ono, "Embedded block coding in JPEG 2000," in *Proc. IEEE Int. Conf. Image Processing*, Vancouver, Canada, Sept. 2000, vol. II, pp. 33-36.
- [70] B.E. Usevitch, "A tutorial on modern lossy wavelet image compression: Foundations of JPEG 2000," *IEEE Signal Processing Mag.*, vol. 18, pp. 22-35, Sept. 2001.
- [71] M. Vetterli, "Wavelets, approximation and compression," *IEEE Signal Processing Mag.*, vol. 18, pp. 59-73, Sept. 2001.
- [72] W3C, PNG (Portable Network Graphics) Specification, Oct. 1996. Available: <http://www.w3.org/TR/REC-png>.
- [73] G.K. Wallace, "The JPEG still picture compression standard," *IEEE Trans. Consumer Electron.*, vol. 38, pp. xviii-xxxiv, Feb. 1992.
- [74] A.B. Watson, G. Yang, J. Solomon, and J. Villasenor, "Visibility of wavelet quantization noise," *IEEE Trans. Image Processing*, vol. 6, pp. 1164-1175, 1997.
- [75] G. Xing, J. Li, S. Li, and Y.-Q. Zhang, "Arbitrarily shaped video object coding by wavelet," in *IEEE Int. Symp. Circuits and Systems (ISCAS 2000)*, vol. III. Geneva, Switzerland, 28-31 May 2000, pp. 535-538.
- [76] W. Zeng, S. Daly, and S. Lei, "Visual optimization tools in JPEG 2000," in *Proc. IEEE Int. Conf. Image Processing (ICIP 2000)*, Vancouver, Canada, 10-13 Sept. 2000, vo. II, pp. 37-40.
- [77] W. Zeng, S. Daly, and S. Lei, "Point-wise extended visual marking for JPEG-2000 image compression," in *Proc. IEEE Int. Conf. Image Processing (ICIP 2000)*, Vancouver, Canada, 10-13 Sept. 2000, vol. I, pp. 657-660.