High-Performance Compression of Visual Information—A Tutorial Review—Part I: Still Pictures

OLIVIER EGGER, PASCAL FLEURY, TOURADJ EBRAHIMI, AND MURAT KUNT, FELLOW, IEEE

Digital images have become an important source of information in the modern world of communication systems. In their raw form, digital images require a tremendous amount of memory. Many research efforts have been devoted to the problem of image compression in the last two decades. Two different compression categories must be distinguished: lossless and lossy. Lossless compression is achieved if no distortion is introduced in the coded image. Applications requiring this type of compression include medical imaging and satellite photography. For applications such as video telephony or multimedia applications, some loss of information is usually tolerated in exchange for a high compression ratio.

In this two-part paper, the major building blocks of image coding schemes are overviewed. Part I covers still image coding, and Part II covers motion picture sequences.

In this first part, still image coding schemes have been classified into predictive, block transform, and multiresolution approaches. Predictive methods are suited to lossless and low-compression applications. Transform-based coding schemes achieve higher compression ratios for lossy compression but suffer from blocking artifacts at high-compression ratios. Multiresolution approaches are suited for lossy as well for lossless compression. At lossy high-compression ratios, the typical artifact visible in the reconstructed images is the ringing effect.

New applications in a multimedia environment drove the need for new functionalities of the image coding schemes. For that purpose, second-generation coding techniques segment the image into semantically meaningful parts. Therefore, parts of these methods have been adapted to work for arbitrarily shaped regions. In order to add another functionality, such as progressive transmission of the information, specific quantization algorithms must be defined. A final step in the compression scheme is achieved by the codeword assignment.

Finally, coding results are presented which compare state-of-the-art techniques for lossy and lossless compression. The different artifacts of each technique are highlighted and discussed. Also, the possibility of progressive transmission is illustrated.

Keywords— *Compression, image processing, JPEG, MPEG, standards, still pictures.*

Manuscript received July 24, 1997; revised March 12, 1999.

O. Egger is with Oasya S.A., 1110 Morges, Switzerland.

P. Fleury, T. Ebrahimi, and M. Kunt are with the Signal Processing Laboratory, Swiss Federal Institute of Technology, 1015 Lausanne, Switzerland.

Publisher Item Identifier S 0018-9219(99)04156-0.

I. INTRODUCTION

Every digital image acquisition system produces pictures in its canonical form. This means that the analog scene is sampled in space and quantized in brightness. If the sampling step size is small enough, the integration ability of the human visual system will give the illusion of a continuous picture to the human observer. In that sense, a digital image is an $N_1 \times N_2$ array of integer numbers. However, this canonical form needs a large number of bits for its representation. For example, a 256×256 picture using 8 bits per pixel needs half a million bits for its representation. The information contained in a sequence of images for video is even higher due to the additional temporal dimension.

Image data compression aims at minimizing the number of bits required to represent an image. Data compression has wide areas of applications. Video telephony between two speakers or teleconferencing on two PC's via the normal twisted-pair phone lines is only possible with very low bit rate compression systems. Nearly all multimedia applications, such as interactive databases (encyclopedias, electronic newspaper, travel information, and so on), need a strong compression of the huge input data consisting of text, audio, and visual information. Other applications can be found in remote sensing, education, and entertainment.

Some applications allow for visible distortions of the input images in exchange for high compression ratios. This is typically the case for applications such as teleconferencing or accessing images from a distant server (for example, applications related to the World Wide Web). Such coding schemes are called lossy. In other applications, however, no distortion of the input image is tolerated. Compression of medical images is a typical example of this category. Coding schemes that introduce no distortion are termed lossless.

In the past, most of the effort has been dedicated to improving the compression ratios of the compression scheme itself. Functionalities of the compression schemes were considered less important. Second-generation image coding techniques [1] attempt to split an image into visual prim-

itives. By choosing these primitives in a semantic way, object-oriented coding schemes can be developed. These schemes compress each object of a scene independently of the others. Their main advantage is that complete interactivity with the scene is possible. For example, different bit rates can be attributed to different objects of the scene. Typically one wants good quality for foreground objects and considers the background less important. For this purpose, coding schemes allowing the compression of arbitrarily shaped objects have to be designed.

Another important functionality for compression schemes is the possibility of progressive transmission or progressive decoding of the bit stream. A typical application is data browsing. A picture has been compressed at a good quality (perhaps even lossless) and the user wants to visualize the picture at a lower quality to save transmission time. This is only possible if the stored bit stream is embedded in the sense that one can reconstruct a coarse version of the image using only a portion of the entire bit stream.

High compression image coders can be split into two distinct groups. The first group is called waveform coding and consists of transform and subband coding (SBC). The second group, called second-generation techniques, consists of techniques that attempt to describe an image in terms of visually meaningful primitives (contour and texture, for example).

A waveform-based coding system consists of performing a decomposition/transform of the image data, quantization of the transform coefficients, and source coding of the quantized coefficients. The first step transforms the image into another representation, where most of the energy is compacted in a few coefficients. This is achieved by means of a subband transform or a block transform, such as the discrete cosine transform (DCT). At compression factors of about 30–40, block transform techniques produce blocking artifacts. Unfortunately, the human eye is very sensitive to such a distortion, therefore block coders are not appropriate for low bit-rate image coding. On the other hand, the main artifact of SBC, at high compression factors (around 50), is due to the Gibbs phenomenon of linear filters and is called ringing effect. To avoid this artifact, morphological subband decompositions lead to good quality decoded pictures at compression ratios as high as 70–80.

The second group of methods is based on second-generation techniques. They attempt to decompose the data into visual primitives such as contours and textures [1], [2]. One approach is to divide the image into directional primitives as proposed in [3]. Segmentation-based coding techniques [4] extract regions from the image data which are represented by their shape and their textural content. Following similar ideas, sketch-based image coding [5] is based on extracting the contours of an image, namely their geometric and intensity information, resulting in the so-called sketch picture. The texture is then defined by the difference between the original and the sketch image and is coded using waveform coding techniques. An extension of this technique has been proposed by Ran *et al*. [6], [7] and is based on a three-component image model. This technique divides the image into the strong edge, texture, and smooth components. The strong edge component is encoded separately whereas the texture and smooth components are encoded using waveform coding techniques. A solution to find the most important image features has been proposed by Mallat *et al*. [8] using multiscale edges. A double layer technique based on multiscale edges and textures has been proposed in [9]. In general, second-generation techniques become efficient at higher compression ratios (about 50) when compared to other methods.

In this paper, the basic blocks of image coding are overviewed. Image coding is fundamentally carried out in two steps: message extraction and codeword assignment.

The first operation—message extraction—consists of transforming the original image into a stream of message symbols. In general, all message-extraction methods are based on a mapping of the original data onto another representation domain. For that purpose different methods have been studied: predictive methods (Section II); block transform methods (Section III); and multiresolution approaches (Section IV). Parts of these methods have been adapted to object-based second-generation coding techniques and are discussed in Section V. The last operation of message extraction is the quantization and is reviewed in Section VI. After message extraction, a codeword is assigned to the extracted symbols. This operation is discussed in Section VII.

Complete state-of-the-art schemes are described in Section IX. Coding results of all those methods are presented in Section X. General conclusions are drawn in Section XI.

II. Pixel-Space Methods

A. Introduction

The simplest methods for image coding are based on predictive methods. These methods are usually used for lossless image compression. Lossless image coding plays an important role in situations such as medical imaging or satellite photography, where no information loss is allowed during compression. A recent survey of lossless compression techniques is presented in [10]. Part of the review presented in [11] is devoted to lossless image compression techniques with special focus on radiological image compression.

For all types of images, direct coding using an entropy coder does not achieve any considerable degree of compression. As an example, consider the "Lena" image shown in Fig. 1. Using the UNIX implementation of the Lempel–Ziv compression, one obtains a compression factor of 1.19. That clearly demonstrates that some form of prior decorrelation is necessary to achieve an acceptable compression ratio. Therefore, all coding techniques appearing in imaging literature employ a method of decorrelation of the information. One possibility is to use predictive models to achieve this goal. Another possibility is to extract self similarity in the

Fig. 1. Test image "Lena."

image, so as to describe an image region with a simple transformation of another region. This is exploited by so-called fractal coding.

B. Predictive Methods

In predictive coding schemes, the next input is predicted based on the digitally coded past. The simplest form of prediction can be achieved by using differential methods. The idea behind differential methods is to encode the value of the difference between the previously encoded pixel and the current pixel. Due to the correlation existing in a natural image, the resulting values to be encoded have a lower dynamic range than the original values.

In predictive schemes one tries to predict a value $x[n_1, n_2]$ based on the coded past. The resulting difference between the original value $x[n_1, n_2]$ and the predicted value $\hat{x}[n_1, n_2]$ is called the prediction error $e[n_1, n_2]$

$$e[n_1, n_2] = x[n_1, n_2] - \hat{x}[n_1, n_2]. \tag{1}$$

The successive values of $e(k, l)$ are then quantized and compressed. If lossless compression is required, the signal $e(k, l)$ must have a limited number of possible values and is not quantized.

The lossless mode of the international Joint Photographic Expert Group (JPEG) standard [12] is a predictive scheme. Seven different prediction methods have been defined in the standard JPEG. All seven are based on a prediction of the next input with up to three previously encoded local neighbors. None of these prediction methods clearly outperform the other ones for all applications. Other predictive lossless compression schemes can be found in literature [13]–[16].

C. Fractal Compression

Iterated functions systems (IFS) theory, closely related to fractal geometry, has recently found an interesting application to image-compression purposes. Barnsley [17] and Jacquin [18] pioneered the field followed by numerous contributions [19], [20]. The approach consists in expressing an image as the attractor of a contractive functions system which can be retrieved simply by iterating the set of functions starting from any initial arbitrary image. The form of redundancy exploited is named piecewise self transformability. This term refers to a property that each segment of an image can be properly expressed as a simple transformation of another part of higher resolution. Then, only the sequence of transformation is sufficient as the description of the full picture. IFS-based still image compression techniques can provide very good performances at high compression ratios (about 70–80) as proved by [21], [22]. Results for fractal compression are shown in Section X.

III. Transform Coding

A. Introduction

Transform coding, also called block quantization, is a widely used technique in practice. A block of data is unitarily transformed so that a large fraction of its energy is packed in relatively few transform coefficients, which are then quantized independently. The optimum transform coder is defined as the one that minimizes the mean-square distortion of the reconstructed data for a given quantization. This turns out to be the Karhunen–Lòeve (KL) transform.

B. The Optimum Transform Coder

Suppose an $N \times 1$ random vector $\underline{x}$ with zero mean and covariance matrix $\boldsymbol{R}$ is linearly transformed by an $N \times N$ matrix $\boldsymbol{A}$ (complex) to produce a vector $\underline{y}$ such that its components are mutually uncorrelated. The quantization of $\underline{y}$ produces then an output $\underline{\hat{y}}$, which is linearly transformed by a matrix $\boldsymbol{B}$ to yield the reconstructed signal $\underline{\hat{x}}$. The problem is to find the optimum matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ and the optimum quantizers such that the overall average mean square distortion is minimized. The solution to this problem [23] can be summarized in the following way.

1) The optimal matrix $\boldsymbol{A}$ is the (KL) [24], [25] transform of $\underline{x}$. More precisely, the rows of $\boldsymbol{A}$ are the ortho-normalized eigenvectors of the auto-covariance matrix $\boldsymbol{R}$.
2) For an arbitrary quantizer, the optimal reconstruction matrix $\boldsymbol{B}$ is given by
$$\boldsymbol{B} = \boldsymbol{A}^{-1}\boldsymbol{\Lambda} \tag{2}$$
where $\boldsymbol{\Lambda}$, modeling the quantization, is a diagonal matrix whose elements λ_i are
$$\lambda_i = \frac{E[y_i \hat{y}_i^*]}{E[|\hat{y}_i|^2]}. \tag{3}$$
3) The optimal Lloyd–Max quantizer minimizes the overall mean square distortion for every element $y(i)$ yielding
$$\boldsymbol{\Lambda} = \boldsymbol{I} \tag{4}$$
and consequently
$$\boldsymbol{B} = \boldsymbol{A}^{-1}. \tag{5}$$

The application of the KL transform to images is quite slow. The KL transform depends on the second-order statistics as well as the size of the image and, hence, the basis vectors are not known analytically. After having computed the transform matrix $\boldsymbol{A}$, the remaining operations for performing the transformation is still quite large for images. There

exists a fast KL transform algorithm [26] only for certain statistical image models. Therefore, the KL transform is not appropriate for image coding applications. There exist numerous transforms which have been applied to image coding. Fortunately, there exists a unitary transform which performs nearly as well as the KL transform on natural images. The mentioned transform is called the DCT and is the subject of the next section.

C. DCT

The one-dimensional (1-D) DCT $X[k]$ of a sequence $x[n]$ is defined as

$$X[k] = \alpha_k \sum_{n=0}^{N-1} x[n] \cos\left(\frac{\pi(2n+1)k}{2N}\right) \quad k = 0, 1, \ldots, N-1 \tag{6}$$

where

$$\alpha_0 = \sqrt{\frac{1}{N}} \quad \alpha[k] = \sqrt{\frac{2}{N}} \qquad k = 1, \ldots, N-1. \tag{7}$$

The inverse transform is then given by

$$x[n] = \sum_{k=0}^{N-1} \alpha_k X[k] \cos\left(\frac{\pi(2n+1)k}{2N}\right) \quad n = 0, 1, \ldots, N-1. \tag{8}$$

The DCT has some very interesting properties. First, it can be noted that the transform kernel is a real function. This is important for coding purposes because only the real part of the transform coefficients of a natural image must be coded. Moreover, there exist fast algorithms for computing the DCT in one or two dimensions [27], [28]. The cosine transform of an $N \times 1$ vector can be performed in $O(N \log_2 N)$ operations via a N-point fast Fourier transform (FFT).

The final objective of the transformation is image compression, and, therefore, it is wished to have a good energy compaction of the transform coefficients. In practice, it is verified that the DCT is very close—in terms of energy compaction—to the optimal KL transform. This has the following reason. The basis vectors of the cosine transform are the eigenvectors of the symmetric tridiagonal matrix $\boldsymbol{Q}$, defined as [23]

$$\boldsymbol{Q} = \begin{pmatrix} 1 & -\alpha & & & \boldsymbol{O} \\ -\alpha & 1 & -\alpha & & \\ & \ddots & \ddots & \ddots & \\ & & -\alpha & 1 & -\alpha \\ \boldsymbol{O} & & & -\alpha & 1 \end{pmatrix} \tag{9}$$

where α is a parameter. Assume the input signal is a first-order stationary Markov sequence, whose covariance function is given by

$$r[n] = \rho^{|n|} \quad |\rho| < 1 \quad \forall n. \tag{10}$$

Then, the inverse of the covariance matrix will be given by

$$\boldsymbol{R}^{-1} = \beta^{-2} \begin{pmatrix} 1 & -\rho\alpha & & & \boldsymbol{O} \\ -\rho\alpha & 1 & -\rho\alpha & & \\ & \ddots & \ddots & \ddots & \\ & & -\rho\alpha & 1 & -\rho\alpha \\ \boldsymbol{O} & & & -\rho\alpha & 1 \end{pmatrix} \tag{11}$$

where $\beta = (1-\rho^2)/(1+\rho^2)$ and $\alpha = \rho/(1+\rho^2)$. Hence, the following approximation is valid:

$$\beta^2 \boldsymbol{R}^{-1} = \boldsymbol{Q} \quad \text{for } \rho \to 1. \tag{12}$$

Consequently, the eigenvectors of $\boldsymbol{R}$ and the eigenvectors of $\boldsymbol{Q}$ will be quite close for a first-order Markov sequence with $\rho \to 1$. A natural image is a highly correlated signal. It is clear that a first-order model of a natural image does not take into account the nonstationarity of the signal such as edges. However, it gives a first approximation. In general, an estimation of the first-order correlation ρ of natural images gives a value around $\rho \approx 0.95$ [29]. Hence, we can assume that the DCT of a natural image is quite close to the optimal KL transform.

This property of the DCT together with the fact that there are fast algorithms have made it a popular transform for image-coding purposes.

D. Block Partitioning

In practice, the application of the transform is not applied to the image itself. The image is divided into $N \times N$ blocks. Each block is then transformed and coded independently. This block processing allows for an increase in compression performance. From simulation results, it is shown that for natural images N is optimal around $8, \ldots, 16$. In the international standard JPEG, a value of $N = 8$ has been chosen.

This block processing has a significant drawback, however, since it introduces a distortion termed blocking effect, which becomes visible at high compression ratios. It is especially visible on image regions with low local variance. This will be illustrated in Section X.

IV. Multiresolution Approaches

A. Introduction

In recent years, SBC of images has become a domain of intensive research. In such a scheme, subbands are computed by filtering the input image with a set of bandpass filters and decimating the results. The subbands each represent a particular portion of the frequency spectrum of the image. The power of this technique resides in its capability to code each subband separately with a bit rate that matches the visual importance of that subband. SBC leads to visually pleasing image reconstruction and does not produce blocking artifacts. In addition, it allows a progressive multiresolution transmission.

The concept of subband decomposition was introduced first by Crochiere [30] in the context of speech coding.

Then Smith and Barnwell [31] solved the problem of perfect reconstruction filter banks for a 1-D multirate system. Following these studies, substantial research effort has been devoted to perfect reconstruction filter banks theory [32]–[34], which was then extended to the two dimensional (2-D) case by Vetterli [35]. Applications of SBC to images was introduced by Woods and O'Neil [36] by means of 2-D separable quadrature mirror filter (QMF) banks.

SBC consists of the following three steps: 1) subband decomposition; 2) quantization; and 3) entropy coding of the subbands. The decoding process involves the inverse of each of these steps to reconstruct the signal.

Several methods have been proposed in the last decade for designing filter banks. The most well-known filters are the QMF's introduced by Johnston [37]. They are two-band filter banks. The design process is based on the minimization of a weighted sum of the reconstruction error and the stop band energy of each filter. QMF's are not perfect reconstruction filters, but they have a linear phase. An alternative solution is the two-band filters proposed by Smith and Barnwell [31], called conjugate quadrature filters (CQF), which allow for perfect reconstruction but have a nonlinear phase. Vaidyanathan *et al.* [33] solved the design problem of M-band perfect reconstruction filter banks. They use perfect reconstruction blocks based on lossless polyphase matrices and optimize adequate parameters. Nayebi *et al.* [38] have developed a framework based on temporal analysis. A design technique leading to numerically perfect reconstruction filter banks has been developed as well by Nayebi *et al.* [39]. This technique is very flexible and addresses the inclusion of additional constraints for specific applications such as low delay, linear phase, high regularity, and so on. Although the above techniques exist for designing M-band and/or perfect reconstruction filter banks, the QMF's designed by Johnston are still the most cited and utilized in the image-coding community. This is partly due to the simplicity of the design technique and the published tables of filter coefficients. On the other hand, filters proposed in [39] and in [40] are relatively long, and hence, they are not suitable for image-coding applications.

Image-coding applications require filter banks with specific features that differ from the classical perfect reconstruction problem or from the design of filter banks for other applications. Psychovisual properties of the human visual system and typical spectral characteristics of natural images have to be taken into consideration. Images are highly nonstationary sources. In general, they are composed of large homogeneous regions and edges which have a small spatial support. Typical natural images have a highly asymmetrical power spectrum with respect to $\pi/2$. A good model is a spectrum proportional to f^{-2} [41], with f being the frequency. Filter banks that take into account the statistics of natural images and reduce the ringing effect have been proposed by Caglar *et al.* [29], [42].

One advantage of SBC of images over block transform coding such as the DCT is the absence of the blocking effect. However, one major artifact is still remaining. It is the ringing effect which occurs around high-contrast edges due to the Gibbs phenomenon of linear filters. This artifact can be reduced or even removed by an appropriate design of the filter bank. Illustrations of this ringing effect will be found in Section X.

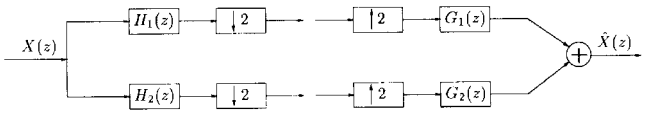


Fig. 2. Two-band analysis/synthesis system. $H_1(z)$ is the lowpass analysis filter, $H_2(z)$ is the highpass analysis filter, $G_1(z)$ is the lowpass synthesis filter, and $G_2(z)$ is the highpass synthesis filter.

For compression purposes, it is important to exploit the existing zero-correlation across the subbands. One powerful approach is the embedded zerotree wavelet (EZW) algorithm proposed in [43]. The EZW algorithm is based on successive-approximation quantization which leads to a completely embedded bit stream. That is, the bit stream of the compressed image can be progressively decoded. This feature allows for important functionalities such as, for example, image browsing. The proposed techniques for noise reduction can be incorporated into the EZW algorithm.

B. Linear Subband Decomposition

1) Analysis/Synthesis Block: Subband decomposition divides the input signal into different subbands. The choice of the filters is an important issue. It has been shown [29], [44], [45] that the filters represent an important factor in the performance of the decomposition for compression purposes.

Fig. 2 shows a two-band filter bank. The filters $H_1(z)$ and $H_2(z)$ are the analysis lowpass and highpass filters, respectively, while $G_1(z)$ and $G_2(z)$ are the synthesis filters. In this system, the input/output relationship is given by

$$\hat{X}(z) = X(z)T(z) + X(-z)A(z) \tag{13}$$

where

$$T(z) = \tfrac{1}{2}\Big[H_1(z)G_1(z) + H_2(z)G_2(z)\Big] \tag{14}$$

and

$$A(z) = \tfrac{1}{2}\Big[H_1(-z)G_1(z) + H_2(-z)G_2(z)\Big]. \tag{15}$$

Perfect reconstruction can be achieved by removing the aliasing distortion $A(z)$ and imposing the transfer function $T(z)$ to be a pure delay of the form $T(z) = z^{-\delta}$, where δ is the delay of the system. By choosing the synthesis filters as $G_1(z) = H_2(-z)$ and $G_2(z) = -H_1(-z)$ the aliasing component is removed, and therefore perfect reconstruction is achievable. Under these constraints the system transfer function becomes

$$T(z) = F(z) - F(-z) \tag{16}$$

where $F(z) = H_1(z)H_2(-z)$ is called the product filter. Now perfect reconstruction is obtained when the product

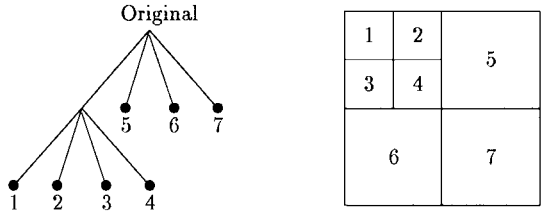


Fig. 3. Illustration of a wavelet decomposition of depth two.

filter $f(k) = \mathcal{Z}^{-1}[F(z)], \mathcal{Z}^{-1}(\cdot)$ being the inverse z-transform, is a power-complementary half-band filter. This means that every odd sample, except one sample $f(\delta)$ is equal to zero, that is

$$f[2n+1] = \begin{cases} 1/2, & 2n+1=\delta \\ 0, & \text{otherwise} \end{cases} \quad n = 0,1,2,\ldots,\frac{L-1}{2} \tag{17}$$

where L is the length of the product filter $F(z)$. Even samples of the product filter f do not affect the perfect reconstruction property. They can be chosen arbitrarily. They are usually chosen to design the product filter as a lowpass filter.

Research on the development of M-band filter banks has been reported in literature in [33] and [39].

2) Tree-Structured Systems: Two-band systems are the basic component of most subband decomposition schemes. They are used in a complete tree structure to define a fine frequency partitioning. As mentioned before, a good model of a natural image is based on a power spectrum proportional to f^{-2} [41], with f being the frequency. That means that most of the frequency is concentrated in low-frequency regions. A suited partitioning of the frequency should therefore have a finer division in low-frequency regions and a coarser partitioning in high-frequency regions. This is the reasoning behind the wavelet decomposition which is the most popular tree-structure in practice. It is defined as follows. At each step of the decomposition, the low-frequency subband is decomposed into its four constituting subbands by ways of of the two-band filter bank applied to the lines and the columns. This procedure is iterated until the lowest subband is smaller than a certain threshold. An example of such a decomposition is shown in Fig. 3.

C. Nonlinear Subband Decompositions

In the previous section, linear subband decompositions have been discussed. It is shown that it is possible to design filters allowing for critical subsampling of the filtered images [32]. In such cases, the decomposed image has the same number of pixels as the original image. It has been shown that the major drawback of such a decomposition scheme is the inherent Gibbs phenomenon of linear filters [44]. This produces an annoying ringing artifact when compressing an image with a high compression ratio. The Gibbs phenomenon affects linear filters but not nonlinear filters, such as morphological filters. Therefore, many attempts have been made to generalize linear subband decomposition

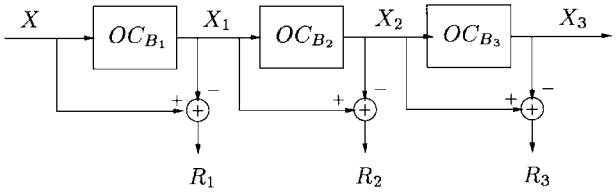


Fig. 4. Classical multiresolution morphological decomposition. The operation OC denotes the morphological open–closing filtering. The decomposition yields a nondetail image and several residual images.

to nonlinear decompositions to overcome the problem of the ringing effect. The most popular class of nonlinear filters used in image processing is that of morphological filters because of their efficiency and simplicity.

Major difficulties in designing morphological filter banks occur because the digital morphological sampling theorem [46] does not provide a tool for designing morphological filters that allow for critical sampling and perfect reconstruction. Indeed, even after a perfect morphological lowpass filter, such as an open–closing filter, there is no way of downsampling the filtered image and reconstructing it perfectly. The problem is that morphological filtering removes small objects without blurring the contours of larger objects. This means that high-frequency information is always contained in the morphologically filtered image. Hence, aliasing always occurs and there is no way of reconstructing the original image perfectly.

Notice that downsampling is a linear process. It is most easily described through a frequency analysis using the sampling theorem [47]; however, the concept of frequency is a strictly linear concept and cannot be used in the context of morphological signal processing.

One important difference between linear and morphological filters is that morphological filters preserve the specificity of the input set. If the input consists of integer valued coefficients only, then the output will also consist of integer valued coefficients. This is not true for linear filters. Using this property, a lossless coding scheme based on the morphological subband decomposition can be defined. The following sections describe these decompositions.

1) Morphological Decomposition: Using mathematical morphology, the analysis of an image is based on object shapes and object sizes. This is in contrast with the frequency concept using linear operations. In terms of mathematical morphology, a multiresolution analysis intends to decompose an image into different subimages, where each subimage contains objects of a specific size. Fig. 4 shows the standard decomposition for multiresolution analysis [48], [49]. This decomposition is a cascade of open–closings which are intended to filter out objects of a certain size at each stage.

The first stage of the decomposition is computed by

$$\begin{aligned} X_1 &= (X \circ B_1) \cdot B_1 \\ R_1 &= X - X_1 \end{aligned} \tag{18}$$

where $(X \circ B_1) \cdot B_1 = OC_{B_1}(X)$ denotes open–closing of X by B_1. R_1 is the first residual image containing only

objects of size smaller or equal than the structuring element B_1. Similarly, the second stage is obtained by

$$X_2 = (X_1 \circ B_2) \cdot B_2$$
$$R_2 = X_1 - X_2 \tag{19}$$

where

$$B_2 = B_1 \oplus B_1 \tag{20}$$

and $\oplus$ is the dilation operator.

Each stage produces a residual image R_i. Together with the last open–closed image X_N the original image can be reconstructed just by adding all together as follows:

$$X = X_N + \sum_{k=1}^{N} R_k. \tag{21}$$

Attempts to use this decomposition for coding purposes are reported in literature [49], [50]; however, the basic problem is the coding of the residual images which are of the same size as the original image. Also, the nondetail image X_N still has a large entropy and is also of the same size as the original image. The only advantage of this decomposition is the absence of ringing effect even under strong quantization. It is clear that a good decomposition for coding purposes must be compact, that is the number of pixels used to represent the original image has to be as close as possible to the number of pixels of the original image. In this case, the representation needs N times more pixels for representing the original and consequently is not appropriate at all for image-compression purposes.

2) Pyramidal Decomposition: Another approach is the pyramidal decomposition [51], shown in Fig. 5. The idea is similar to the previous decomposition. The original image is first open–closed by a structuring element B to give the filtered image X_1

$$X_1 = (X \circ B) \cdot B = OC_B(X) \tag{22}$$

then X_1 is down-sampled by a factor of two in the two directions. Denote the image X_1 by $x_1[n_1, n_2]$ in the spatial domain. The downsampled image $y_1[n_1, n_2]$ is then

$$y_1[n_1, n_2] = x_1[2n_1, 2n_2] \quad n_1 = 0, 1, \ldots, (\tfrac{1}{2} N_1) - 1$$
$$n_2 = 0, 1, \ldots, (\tfrac{1}{2} N_2) - 1. \tag{23}$$

The image Y_1 is then further decomposed at each stage. In order to preserve perfect reconstruction the computation of the residual images has to be performed the following way. The image Y_i is first upsampled by a factor of two in the two directions which gives

$$z_i[n_1, n_2] = \begin{cases} y_i\left[\dfrac{n_1}{2}, \dfrac{n_2}{2}\right], & n_1 = 0, 2, 4, \ldots \\ & n_2 = 0, 2, 4, \ldots \\ -\infty, & \text{otherwise.} \end{cases} \tag{24}$$

The upsampled image is then filtered by a morphological reconstruction filter which should produce an image as close to X_i as possible. The residual image is then computed as

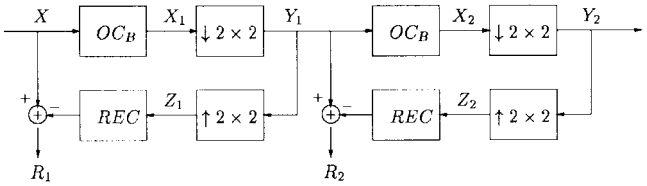


Fig. 5. Pyramidal decomposition using mathematical morphology. The operation OC denotes the morphological open–closing filtering. The decomposition yields a downsampled nondetail image and several residual images.

the difference between the image Y_{i-1} of the previous stage and the reconstructed image, that is

$$R_1 = X - REC(Z_1) \tag{25}$$

for the first stage and

$$R_i = Y_{i-1} - REC(Z_i) \qquad i = 2, 3, \ldots \tag{26}$$

for the next stages. It is clear that the performance of this decomposition depends directly on the performance of the reconstruction filter because the residual images do not only contain the residual objects, which are contained in the difference $Y_{i-1} - X_i$ but also contain the reconstruction error

$$\varepsilon_i = X_i - REC(Z_i). \tag{27}$$

In that sense, the variances of the residual images are higher than with the decomposition given in Section IV-C1. Since we can assume that ε_i and X_i are uncorrelated, we have

$$\sigma_{Ri}^2 = \sigma_{\varepsilon i}^2 + \sigma^2 \tag{28}$$

where σ^2 is the variance of the difference $Y_{i-1} - X_i$. The smaller $\sigma_{\varepsilon i}^2$ the better the decomposition. The advantage of this decomposition is that the residual images are getting smaller by a factor of four at each stage.

Although, such a decomposition is much more appropriate than the multiresolution decomposition, it seems that there is still a waste of space since the decomposed image needs more space than the original image. Optimal would be a compact decomposition which would require the same number of pixels as the original image.

3) Morphological Subband Decomposition: In order to have a tool which can be generalized to the nonlinear case, we will use the concept of a half-band filter. Let us give the definition of a linear half-band filter.

Definition 1: A linear filter is called a half-band filter if every odd sample of its impulse response $f(k)$ is zero, except one sample $f(\delta)$, that is

$$f[2n+1] = \begin{cases} 1/2, & 2n+1 = \delta \\ 0, & \text{otherwise} \end{cases} \quad n = 0, 1, 2, \ldots, \frac{L-1}{2} \tag{29}$$

where L is the length of the half-band filter $f(k)$.

Suppose a signal $x(k)$ is first downsampled and upsampled by a factor of two. Call this signal $y(k)$. Then, it is filtered by a half-band filter $f(k)$. This procedure is shown

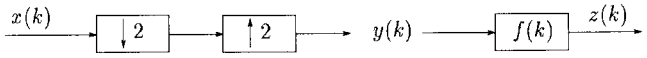


Fig. 6. The signal $x(k)$ is downsampled and upsampled by a factor of two to obtain $y(k)$, which is then filtered by a half-band filter with impulse response $f(k)$ leading to the output $z(k)$.

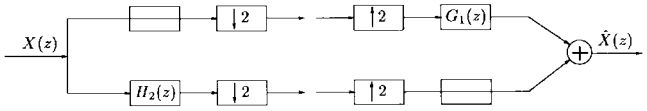


Fig. 7. Filter bank with specific filters yielding perfect reconstruction if $G_1(z)$ is a half-band filter and $H_2(z) = G_1(-z)$.

in Fig. 6. Let us define $z(k)$ as the convolution of $y(k)$ with $f(k)$

$$z(k) = y(k) * f(k). \tag{30}$$

Then, every alternate sample of $z(k)$ is given by

$$z(2k) = \tfrac{1}{2}\, x(2k - \delta) \tag{31}$$

that is, the input is not processed (except a factor of $\frac{1}{2}$) for these samples. Note that (31) implies (29). Moreover, (31) allows us to generalize the concept of a half-band filter to the nonlinear case.

Definition 2: Consider a signal $x(k)$, which is downsampled and then upsampled by a factor of two to give $y(k)$. A nonlinear filter is then called a half-band filter if every alternate sample of the filtered signal $y(k)$ is equal to the corresponding samples in the original signal $x(k)$. Define $\mathcal{F}_y(k)$ the output of the nonlinear half-band filter. It then satisfies

$$\mathcal{F}_y(2k) = c \cdot x(2k - \delta) \tag{32}$$

where c is an arbitrary constant.

Note that such nonlinear filters do exist, like erosion, dilation or median filter, if the region of support is chosen appropriately.

Let us recall that perfect reconstruction of the linear two-band filter bank of Fig. 2 is achieved if the convolution of $h_1(k)$ and $g_1(k)$ is a half-band filter and if the filters respect the bi-orthogonality conditions given by $H_2(z) = G_1(-z)$ and $G_2(z) = -H_1(-z)$ [52]. Now, suppose $H_1(z) = 1$. Then perfect reconstruction is achieved if $G_2(z) = -1, G_1(z)$ is a half-band filter and $H_2(z) = G_1(-z)$. This filter bank is shown in Fig. 7.

Let us replace the filter $G_1(z)$ by a generalized half-band filter $\mathcal{M}(\cdot)$. It is clear that this filter cannot be described with its z-transform since it is a nonlinear filter. However, from the bi-orthogonality condition, in the case of linear filter banks, it is known that $H_2(z) = G_1(-z)$ yields perfect reconstruction. The solution to the problem of finding the corresponding high-pass analysis filter, in the nonlinear case, can be obtained by understanding exactly what it means to negate the variable z of a linear half-band filter. Again, consider the impulse response of a half-band filter. Every other sample is zero, except the median sample.

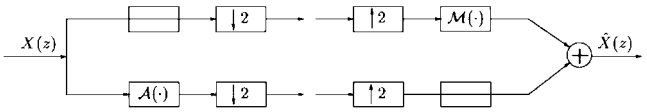


Fig. 8. Filter bank with morphological filters yielding perfect reconstruction if $\mathcal{M}(\cdot)$ is a generalized half-band filter and $\mathcal{A}(\cdot) = \mathcal{I} - \mathcal{M}(\cdot)$.

Now, negating the variable z has the following effect on the impulse response:

$$\mathcal{Z}^{-1}\{F(-z)\} = (-1)^k f(k). \tag{33}$$

Since every odd sample is zero, the negation of the z variable will only have an effect on the middle sample. Hence the following equation holds for a zero-phase ($\delta = 0$) half-band filter:

$$F(-z) = 1 - F(z). \tag{34}$$

This situation can now be generalized to the nonlinear case. Denote $\mathcal{A}(\cdot)$ the nonlinear high-pass analysis filter. Then

$$\mathcal{A} = \mathcal{I} - \mathcal{M} \tag{35}$$

where $\mathcal{I}$ is the identity operation. This nonlinear filter bank yields perfect reconstruction and is shown in Fig. 8. Although this filter bank is a particular case, it invalidates the assertion that morphological filter banks with perfect reconstruction are impossible in general.

The effect of taking the lowpass analysis filter as the identity operation will be to introduce aliasing in the downsampled lowpass subband. However, it is shown that good subband filters, such as the asymmetrical filter banks (AFB's) [44], have a similar property since the lowpass filter has quite a poor frequency response. What matters most for image compression is the behavior of the lowpass synthesis filter, because it is designed mainly to filter the introduced quantization noise, but it is also the main origin of the ringing effect. With the kind of filter bank just discussed we have a complete control on the lowpass synthesis filter. The use of a morphological filter at this stage will completely eliminate the ringing effect. Furthermore, the proposed morphological subband decomposition (MSD) has the desired property of representing the original image by the same number of pixels as the original image.

D. Lossless Decompositions

In contrast to lossy image coding, special care has to be dedicated to the coefficients in the subband domain when lossless compression is intended. Indeed, classical subband/wavelet coefficients are floating point numbers. The output of these filters are therefore floating point numbers as well. This representation is not suitable for subsequent lossless entropy coding. Even if the filter coefficients can be made integers by a single normalization factor, the dynamic range is usually too wide to be effectively coded. Therefore, different multiresolution techniques have to be designed for the lossless compression of images than those used for the lossy compression. Several multiresolution approaches have been presented for the lossless coding of images [53]–[57].

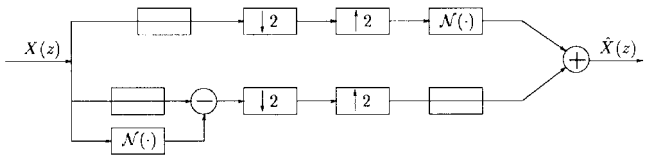


Fig. 9. Filter bank with morphological filters allowing for perfect reconstruction and having only integer valued coefficients. The operator $\mathcal{N}(\cdot)$ denotes $\mathcal{Q}(\mathcal{M})$, where $\mathcal{Q}(\cdot)$ is a quantization function mapping a floating point valued coefficient to an integer valued coefficient.

In the previous section, the MSD has been described as a tool for the compression of images. It has been stated that it does not suffer from any ringing effect at low bit rates. The MSD has another interesting property. All the filtering in the decomposition is performed with morphological filters which have the property of preserving the specificity of the input set (see Section IV-C). This makes it possible to define a lossless compression scheme.

In order to achieve lossless compression of the image, two conditions must be satisfied by the decomposition filters.

1) The filter bank must allow perfect reconstruction. This means that if no quantization is applied to the decomposed coefficients, the reconstructed image must be identical to the original image.
2) The coefficients of the subbands must be representable by a limited number of bits.

As shown in Section IV-C3, the MSD enjoys the property of perfect reconstruction. Besides, median filters on a region of support of size six are very efficient for that purpose [58]. Since this median has an even number of samples, the output of such a filter will no longer yield integer valued coefficients. This is due to the fact that the median value of an even number of samples is defined by the average of the two middle samples.

One possibility of satisfying the second condition is to modify the MSD in the following way. The highpass synthesis filter is defined by the generalized half-band filter $\mathcal{M}(\cdot)$. Let us define a new half-band filter $\mathcal{N}$ by

$$\mathcal{N} = \mathcal{Q}(\mathcal{M}) \tag{36}$$

where $\mathcal{Q}(\cdot)$ is a quantization function mapping a floating point valued coefficient to an integer valued coefficient. The flow diagram of the new decomposition is shown in Fig. 9. Such a filter bank still allows for perfect reconstruction, as the input data are integers.

V. Segmentation-Based Approaches

A. Motivation

New applications in a multiresolution environment result in the need for additional functionalities and requirements for video coding schemes. Most of them are closely related to object-oriented manipulations. Object-oriented functionalities require a prior segmentation of the scene into regions of interest. These regions of interest are the actual objects and are generally of arbitrary shapes. There are two important issues involved in a region-based scheme: 1) the representation and coding of the region shapes and 2) the representation and coding of the interior of the regions. When considering moving picture sequences, the issue of tracking is also raised; this will be covered in Part II of this paper. This section will focus on the representation and coding of the interior of the regions.

An efficient representation of the region interior should provide a good decorrelation of the data in order to obtain a high energy compaction. When compressing rectangular pictures, this decorrelation is performed by using appropriate block transforms such as a DCT or subband/wavelet transforms. Various approaches have been proposed to generalize the block-based techniques to arbitrarily shaped regions.

One approach is to represent the arbitrarily shaped region using a classical block transform such as DCT. An appropriate extension is performed in those blocks containing pixels not belonging to the region interior. The block is filled with pixel values. The whole block is then transformed and coded. Several approaches to perform the extension have been proposed [59], [60]. This approach obviously has the disadvantage of needing to transmit more coefficients than pixels in the region.

A technique using a Gram–Schmidt orthogonalization procedure to find an orthonormal basis inside the region of interest has been proposed by Gilge in [61]. This procedure is computationally expensive both at encoder and decoder sides, as it requires the computation of a different orthonormal basis for each different region.

A technique using an iterative procedure based on the theory of successive projection onto convex sets has been proposed by Chen *et al*. [62]. This technique has proven to be efficient in a rate-distortion (R-D) sense. However, it suffers from two drawbacks: 1) it is computationally heavy due to the involved iteration and 2) if the number of pixels of the shape is the same as the number of coefficients in the transform domain, then perfect reconstruction of the shape is not guaranteed.

The algorithm proposed by Sikora *et al*. [63], [64] performs a shifting of the input pixels in such a way that a DCT can be performed along the shifted lines. This technique, known as SA-DCT (see Section V-C), has become very popular due to its coding efficiency. The main drawback of this method is that it does not perform a full decorrelation of neighboring pixels in a given region.

B. POCS-Based Block Transform (PBT)

The PBT, where POCS stands for projection onto convex sets, is based on an iterative procedure to determine the best transform coefficient values. Two convex sets are defined: one in the spatial domain and one in the frequency domain. One iteration of the PBT (see Fig. 10) will project the input image onto the spatial domain set, transform this projected set into the frequency domain, project the transformed coefficients onto the frequency domain set, and transform

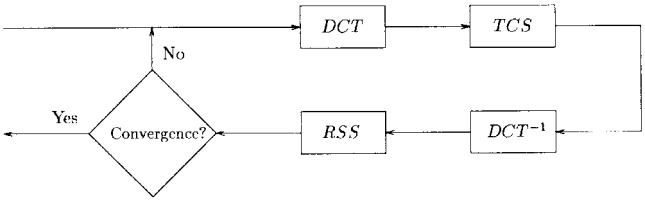


Fig. 10. General flow diagram of the POCS-based block iterative process. The transform coefficient set (TCS) keeps only a subset of the transform coefficients, whereas the region of support set (RSS) keeps only a subset of the spatial coefficients.

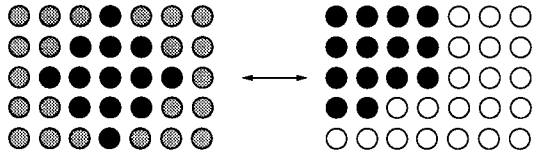


Fig. 11. (a) Region enforcing in the space domain. Black circles: original shape; grey circles: pixels outside the region of interest. (b) Region zeroing in the frequency domain. Black circles: significant transform coefficients; white circles: zeroed transform coefficients.

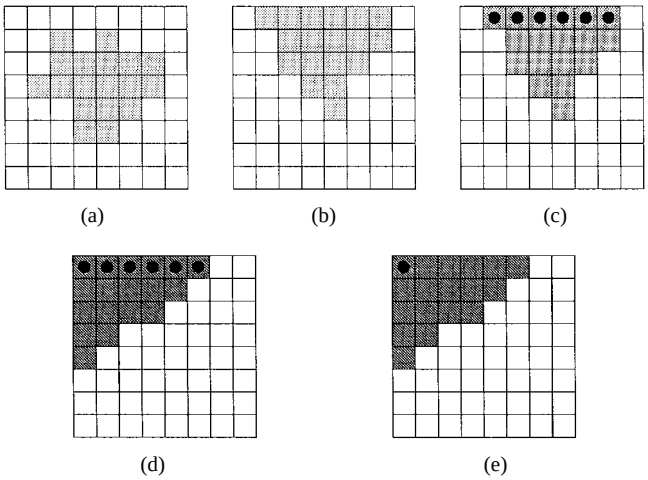


Fig. 12. Illustration of the SADCT. (a) Original shape (grayed). (b) Shape after first shifting of pixels. (c) Shape obtained after performing a DCT_N on all the columns, with the DC coefficients marked with a black circle. (d) Shape after second shifting of pixels. (e) Shape obtained after performing a DCT_N on all the lines, with the DC coefficient marked with a black circle.

the coefficients back to the spatial domain. Therefore, close convex sets have to be defined. They are associated with *a priori* constraints that are based on the characteristics of the input shapes.

The first set is defined based on the theory of transform coding. The goal of a transform for coding purposes is to compact the energy of the input image into few coefficients. Typically, for natural images, only a few coefficients of the DCT transform are important to represent the input image. The set of images which can be represented using a selected group of transform coefficients constitute the first set. This set is termed transform coefficients set (TCS). The projection of the transform coefficients of an input onto this set can be obtained in zeroing all coefficients not in the TCS. This is called region zeroing.

The second set is derived from the fact that the values of the pixels outside the region of interest are irrelevant. This set if referred to as the region of support set (RSS). The projection of an arbitrarily shaped region onto the RSS can be obtained by simply replacing those pixels corresponding to the interior of the region by their original values, and ignoring the other pixels. This procedure is called region enforcing in the space domain.

Both those sets are illustrated in Fig. 11. In Fig. 11(a), one can see the original shape represented by the black circles. The grey circles represent the pixels being irrelevant because they do not belong to the original shape. Applying region enforcing to the region in grey would keep only the black pixels after the projection. In Fig. 11(b), the transform domain is shown. The black circles represent the coefficients which are kept and the white circles represent the coefficients which are zeroed by the region-zeroing process.

One iterates over the two sets until a satisfying result is obtained. Although, the convergence of the algorithm is guaranteed [65], this transform cannot achieve perfect reconstruction. Even though DCT is a compact frequency transform, zeroing out some coefficients in the frequency domain will inevitably results in a loss of information. However, as most of the energy of a natural image is concentrated in the low-frequency coefficients, the loss is minimized.

C. Shape-Adaptive DCT

The shape-adaptive DCT (SADCT) is based on predefined orthogonal sets of DCT basis functions. The basic concept of the method proposed by Sikora [63], [64] is as follows. The original shape is segmented into $N \times N$ blocks with N being eight in most cases. On the blocks where all the pixels belong to the original shape, the classical block DCT is applied. For those blocks where some pixels do not belong to the original shape, the algorithm illustrated in Fig. 12 is performed. The first step consists of shifting all the pixels of the region to the upper bound of the block. As an example, the shifting of all the pixels of the original shape shown in Fig. 12(a) is presented in Fig. 12(b). Then, depending on the number of pixels of each particular column, a DCT transform matrix DCT_N containing a set of N DCT_N basis vectors is selected. The DCT_N transform matrix is defined as

$$\mathrm{DCT}_N(p,k) = c_0 \cdot \cos\left[p \cdot \left(k+\frac{1}{2}\right) \cdot \frac{\pi}{N}\right] \qquad k,p = 0,\ldots,N-1 \tag{37}$$

where $c_0 = (1/\sqrt{2})$ if $p = 0, c_0 = 1$ otherwise, p denoting the pth DCT basis vector. The N vertical DCT coefficients c_j for each column x_j can be computed as

$$c_j = \frac{2}{N}\,\mathrm{DCT}_N \cdot x_j. \tag{38}$$

The result of the transforms for each column is illustrated in Fig. 12(c). The next step consists of shifting all the pixels to the left bound of the block as shown in Fig. 12(d). Then,

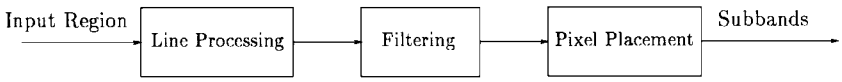


Fig. 13. General flow diagram of the SAWT.

on all the lines, the DCT_N transform is applied to obtain the final transform coefficients, illustrated in Fig. 12(e).

This algorithm presents the advantage that the final number of transform coefficients is the same as the number of total pixels of the original shape. This transform also enjoys the property of being reversible if the original segmentation is known. Due to its computational simplicity and its effectiveness, it is the shape-adaptive transform having the biggest popularity. The main drawback of this approach lies in the decorrelation of nonneighboring pixels. Also, since this approach is block-based, blocking artifacts will occur at high compression factors.

D. Subband Transform for Arbitrarily Shaped Regions

1) General Requirements: In this section, an algorithm to perform arbitrarily shaped transforms is examined. As mentioned before, it is crucial to have a decomposition satisfying the following two conditions.

1) The transformed shape must be represented by the same number of coefficients as present in the original shape.
2) The filtering must be performed on pixels which are neighbors of each other in the original shape.

The first condition is a requirement necessary in order to have a good coding performance. Decompositions which do not fulfill this condition show a poorer coding performance. It has to be noticed that if the goal of the decomposition is not coding but a multiresolution analysis, then this condition is not required anymore.

The second point is important in order to have a full decorrelation in the transform domain. As seen for the SA-DCT in Section V-C, pixel neighborhood is not preserved and artificial contours are created. This leads to an increased difficulty in decorrelating the information and thus increases the variance in the highpass subbands.

2) Shape-Adaptive Wavelet Transform: The proposed shape-adaptive wavelet transform (SAWT) is performed in a separable way as in conventional decomposition schemes. The transformation is done in three steps, as illustrated in Fig. 13. The first step consists in processing each line such that no hole is present anymore. The second step decomposes the processed line into N subbands while the last step places the filtered wavelet coefficients in their right locations.

The line processing step writes for each line all pixels belonging to the region in a vector sequentially. This procedure eliminates all holes present in the original shape and is shown in Fig. 14. This operation satisfies condition 1) of the previous section. Indeed, neighboring pixels remain neighboring pixels after the processing; however,

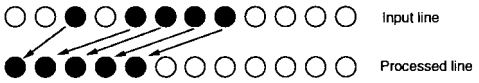


Fig. 14. Processing of each line before filtering.

it is to be noted that holes present in the original shape will produce artificial contours as well. Note that this drawback is also to be found in the SADCT. There is one major reason why this artifact may not degrade the performance of the technique. The objects are usually segmented in such a way that they are spatially compact. This leads to objects where holes are not very frequent.

The processed vector is then decomposed into N bands by means of a filter bank in a second step. A filter bank requires the input to have a size being a multiple of the number of samples. Since the transform must be operational on arbitrarily shaped regions, one cannot ensure that this condition will be fulfilled. In order to overcome this problem, these extra pixels are not processed in the filter bank but are put in the lowpass subband after having been multiplied by the gain of the lowpass filter. This is a reasonable choice since the statistics of the lowpass filter are very similar to the statistics of the original signal. It is important not to create a change of statistics for these extra pixels. Since the gain of the lowpass filter is in general not equal to one (usually $\sqrt{2}$), one has to multiply all these extra pixels by this gain. The usual number of bands for wavelet decompositions is $N = 2$. In that case, there is a maximum of one extra pixel per line. This procedure is illustrated in Fig. 15 for a two-band filter bank. The upper left picture shows the original shape. The upper right figure shows each line after all the holes have been filled. These lines are then filtered with the two-band filter bank shown in the lower right part of the figure. Finally, the extra pixels represented by the grey circles are put in the lowpass subband as illustrated in the lower left part of the figure. It is important to mention that linear phase filters are appropriate for this decomposition because they allow a symmetric extension of each line. This reduces greatly the border effects which could be introduced otherwise. Note that asymmetrical filter banks satisfy this requirement.

In a third step, the filtered samples have to be placed into subbands such that each subband preserves the original shape of the region. This is important for the purpose of obtaining a maximum decorrelation of the filter bank. For every group of N pixels of the original region the samples are distributed into the N subbands such that the number of pixels in each subband does not differ by more than $N - 1$. As illustrated in Fig. 16, this procedure ensures that the subbands are very near in shape to the original region scaled by a factor of N. An example of a wavelet

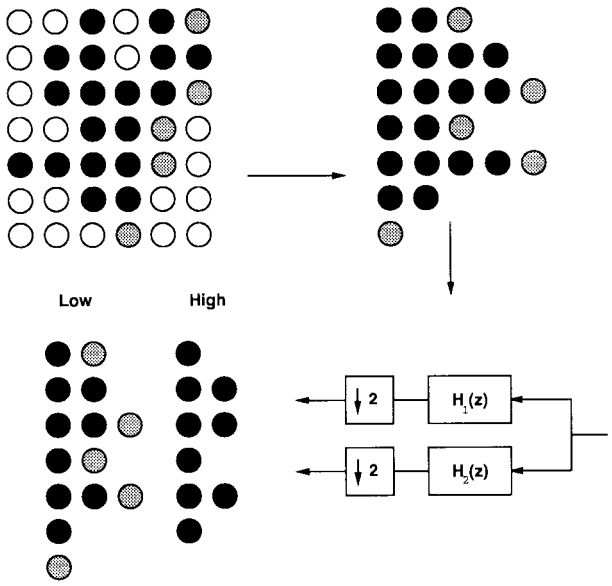


Fig. 15. Decomposition of a region into two subbands. Black circles are pixels belonging to the region. Grey circles represent extra pixels which are not processed by the filter.

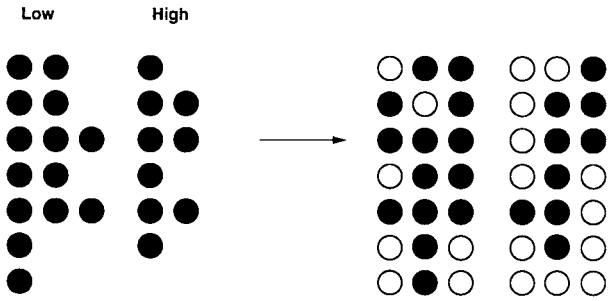


Fig. 16. Placing of the decomposed samples in the two subbands. (a) Subbands obtained after the filtering. (b) Each pixel is placed at the correct location in the subbands.

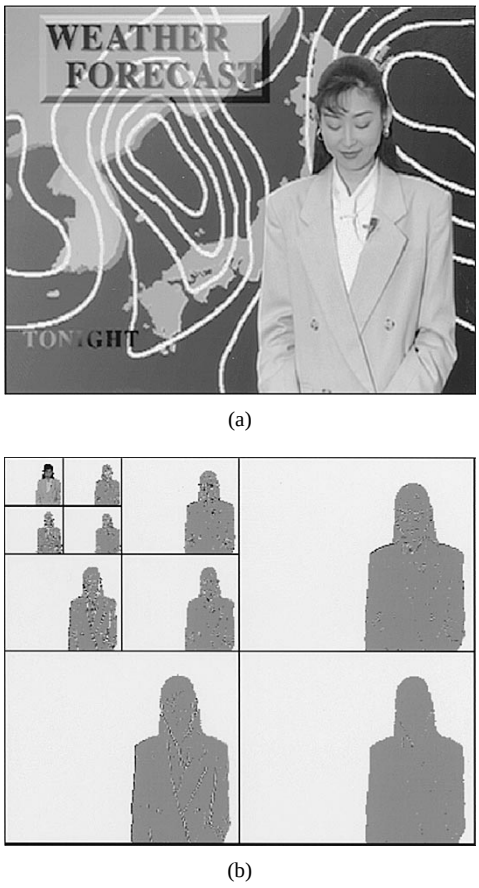


Fig. 17. Example of an SAWT on the picture "Weather." (a) Shape to be coded: foreground person. (b) Resulting subbands after performing the SAWT.

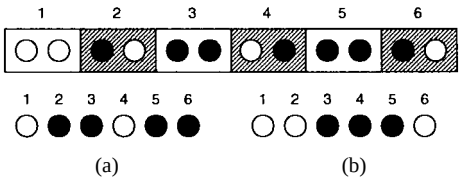


Fig. 18. Illustration of the placement of the decomposed samples in the two subbands. (Top) Line to be decomposed. The dashed group of two pixels are those having only one pixel belonging to the region of interest. (Bottom) Resulting placement of the pixels in (a) the lowpass subband and (b) the highpass subband such that they are evenly distributed.

decomposition of a shape is shown in Fig. 17. One can clearly observe that the subbands are very near in shape to the original region.

This algorithm is not reversible in the sense that in general it is not possible to find the original contour of the region from the contours of the subband shapes. This is due to the fact that one imposes an equal number of pixels in each subband of the same level (if we except the mentioned extra pixels). Hence, all pixels belonging to a group of N pixels with some of those pixels not belonging to the region are evenly distributed among the subbands. This is illustrated for $N = 2$ in Fig. 18. In the figure, the dashed group of two pixels are those having only one pixel belonging to the region of interest. At the appearance of such a group, one pixel is attributed to the lowpass subband. At the next appearance it is given to the highpass subband. In Fig. 18, bins number 2, 4, and 6 have this structure of only one pixel belonging to the region. Hence, pixels number 2 and 6 will be present in the lowpass subband while pixel 4 will be in the highpass subband.

If at a specific location in the subbands there is a pixel in the lowpass subband but not in the highpass subband, it is impossible to know if in the reconstructed shape it is the first or the second pixel that belongs to the region. This is illustrated in Fig. 19. Fig. 19(a) shows the lowpass subband and the highpass subband. For pixels 2, 4, and 6, it is impossible to retrieve the correct constellation of the original shape. There are always two possibilities, and these are shown in Fig. 19(b).

Although one cannot reconstruct the original shape from the subband shapes, we do not have to send additional information since the shape is coded in a different channel anyway. The procedure to decompress a region coded

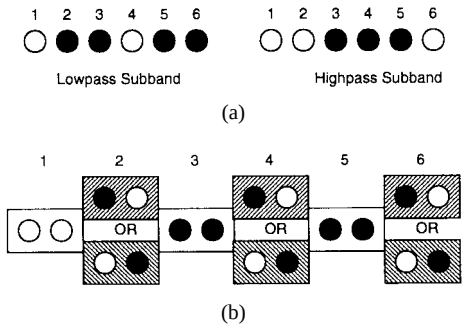


Fig. 19. From the samples in the two subbands, it is impossible to retrieve to original shape. (a) For pixels 2, 4, and 6, it is impossible to retrieve the correct constellation of the original shape. (b) There are always two possibilities.

with the SAWT algorithm must be done in the following chronology:

1) decoding of the contour information;
2) decomposition of the shape into the subband shapes by using the SAWT without the filtering;
3) reconstruction of the original shape and its content by applying the inverse filter bank of the SAWT.

E. The Shape-Adaptive Transform for the Nonlinear Transforms

In the previous section, an algorithm has been proposed to perform a subband decomposition on arbitrarily shaped regions. This algorithm is based on filtering each line independently. Therefore, it works properly only for separable filter banks with monodimensional subband filters. However, some nonlinear subband transforms, such as the MSD, use filters with a 2-D region of support.

In the case of the proposed nonlinear filter banks, having to deal with a degenerated filter bank simplifies the task of defining a shape-adaptive version of the transform. The lowpass analysis filter is not present in the filter bank since the lowpass subband is defined as a downscaled version of the original subband. Hence, one can define the lowpass subband of the shape-adaptive transform as the downscaled version of the original shape. The procedure to obtain this subband is as follows. Fig. 20 illustrates the case $N = 2$, i.e., two subbands. Each line of the original shape is downsampled by a factor of two. Hence, every odd pixel of the original shape is contained in the lowpass subband. If an odd pixel belongs to the original shape, it will belong to the subband shape as well. On the other hand, if this odd pixel did not belong to the original shape, it will not belong to the subband shape either. Fig. 20(a) shows the original shape; black circles belong to the shape, white circles do not belong to the shape, and the dashed rectangles highlight the pixels belonging to odd columns. The downsampled version of the shape is represented in Fig. 20(b).

The highpass subband is composed of the difference between the original image and the image filtered with the generalized half-band filter. One can define a similar procedure for arbitrarily shaped regions. Suppose the generalized

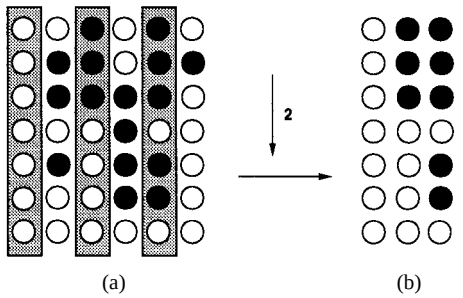


Fig. 20. Downsampling of the original shape to obtain the lowpass subband. (a) Original shape. Black circles belong to the original shape. White circles do not belong to the original shape. The dashed rectangles highlight the pixels belonging to odd columns. (b) Downsampled version of the shape, corresponding to the odd columns.

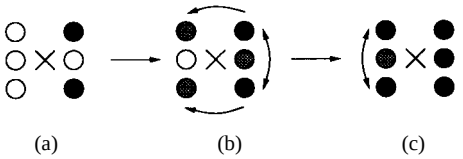
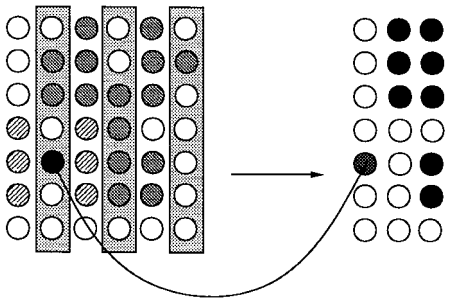


Fig. 21. Example of the iterative procedure to fill all the region of support of the generalized half-band filter. (a) Region of support of six pixels with only two pixels belonging to the shape. (b) After the first iteration, the three grey pixels can be computed. (c) After the second iteration, the last (grey) pixel can be computed.

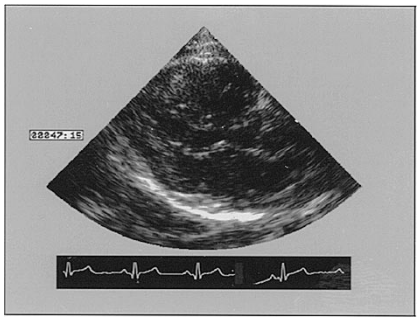
half-band filter has a region of support of N samples. In the case of rectangular images, it is guaranteed that for all the N samples there is a pixel to be filtered. The only exception is on the border of the picture. This can be overcome easily by applying a symmetric extension of the image. In the case of arbitrarily shaped regions there are three cases of interest.

1) All N pixels of the region of support of the filter belong to the region. In this case, the same filtering procedure is applied as for rectangular images.
2) There are only m pixels of the region of support of the filter belonging to the original shape, where $0 < m < N$. In this case, a successive symmetrical extension of the region of support is applied until all N samples have been extended. This procedure is illustrated in Fig. 21. Fig. 21(a) shows a region of support of six pixels for the nonlinear filter. Only two pixels (black circles) belong to the shape. On the first iteration, three more pixels can be computed by taking the average values of the neighbor pixels illustrated by the arrows in the figure. The second iteration is able to compute the missing pixel again by taking the average of its two neighbors.
3) No pixels of the region of support of the filter belong to the original shape. In this case, the output pixel is not filtered but added to the lowpass subband. This procedure is illustrated in Fig. 22. The black circle on the left part of the figure has no neighbors belonging to the shape. All pixels on the region of support of the filter (dashed circles) do not belong to the original shape. Thus, it is put in the lowpass subband.

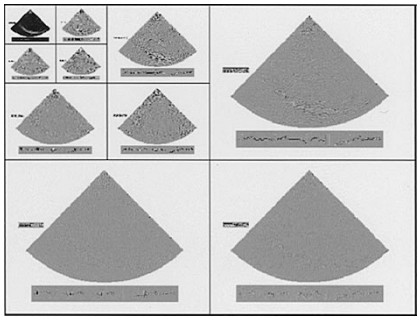


(a) (b)

Fig. 22. (a) The shape is represented by the dark grey circles and the black circle. The rectangles highlight the odd columns. The pixel represented by the black circle cannot be filtered because no pixel of the region of support (dashed circles) belongs to the original shape. (b) Thus, this pixel is put into the lowpass subband at the corresponding place.



(a)



(b)

Fig. 23. Example of an SAND on a ultrasound picture. (a) Shape to be coded. (b) Resulting subbands after performing the SAND.

This definition for the shape-adaptive nonlinear decompositions (SAND) allows for perfect reconstruction of the original shape. The comments on the subband shape of the SAWT also apply to this decomposition. An example of a decomposition of a shape is shown in Fig. 23. One can clearly observe that the subbands are very close in shape to the original region.

F. Shape Coding

In this section, we briefly present major shape-coding techniques used in literature. The user is referred to [66] for a more detailed overview. The emphasis will be given to binary shape coding as opposed to alpha shape coding, which has been recently introduced in a number of applications using concepts from computer graphics in order to represent partially transparent object layers. However, it is possible to apply similar principles to code alpha plane shapes.

Region shape representation has been investigated in the past for arbitrary and constrained shaped regions [67], [68]. They can be classified in three categories: bitmap; intrinsic; and contour-based techniques.

Designed only for compression, bitmap-based techniques apply binary image coding methods, such as those developed for facsimile transmission, to shape images. Two typical bitmap coding schemes are the modified-modified read (MMR) [69], [70] and the context-based arithmetic encoding (CAE) [71]. The former has been used successfully in the facsimile group four transmission norm, whereas the latter has been included in the Motion Picture Experts Group (MPEG)-4 video compression norm. These methods are of particular interest when shape coding is integrated in a block-based scheme as in current still image coding standards, as they are easily modified to accommodate for the fixed segmentation [72].

Intrinsic shape representation considers the solid shape rather than its boundaries. In its most trivial form, an intrinsic shape coding technique is a bilevel coding of inside/outside binary information of an object. This is performed in decomposing the shape into smaller simple elements, by means of quadtree-based techniques [68] or representing it by its skeleton [73], [74]. The inverse decomposition is then applied to the transmitted simple element to recover the shape. Some methods may present characteristics like progressive or lossy shape coding. Current object-based standards like MPEG-4 use this approach due to their good compression efficiency and their simplicity. However, such an approach lacks a semantic description of objects, which is of interest in applications where high-level information is needed (content-based search, image understanding, pattern matching, etc.)

Contour-based techniques use a transform to convert the object into contours representing its outline. An inverse transform recovers the shape from this contour representation. This transformation should preserve information to allow lossless shape coding using contours. When simplification hypotheses are available, the transformation can often be expressed in a very compact form.

The key part in contour-based techniques is the actual coding of the contour information. Several connectivity methods can be used to define what relative positions two neighboring contour pixels can take. The most common neighboring descriptions are, as shown in Fig. 24, four connectivity, eight connectivity, and six connectivity. Let i and j be two neighboring contour pixels. In a four-connected contour, pixel i is either to the north, east, south, or west of pixel j, and in a eight-connected contour it can also be to the northeast, southeast, southwest, or northwest, in addition to the previous directions. A third connectivity scheme represents six-connected contour pixels. As opposed to four- and eight-connected contours that lie on pixels, six-connected contours lie in between pixels. The line between two pixels is also called crack edge [66]. This last possibility is a more natural representation; however,

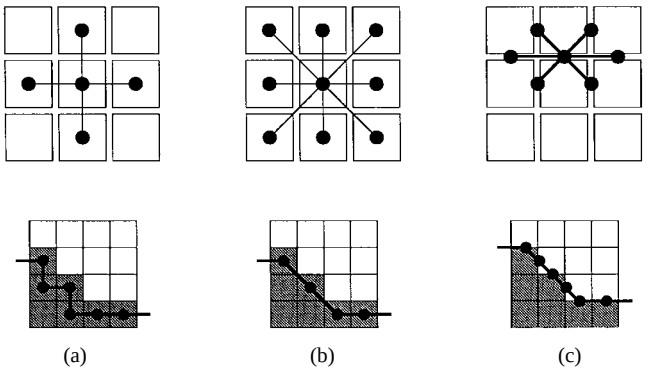


Fig. 24. The most common connectivities, along with an example of the shape representation in: (a) four connectivity; (b) eight connectivity; and (c) six connectivity. Note that six connectivity is defined on the border of the pixels (crack edges), whereas the other two are defined in the middle of the pixel.

the corresponding grid implementation is more complex when compared to the other representations.

A contour tracking algorithm then extracts lists of connected contour pixels. In most cases, the corresponding shape boundaries are described by a closed contour. However, depending on the selected connectivity and the existence of thin details, or borders, some parts of the contour may remain open. Two solutions exist to handle that case.

The first possibility is to let the contour tracker move backward. The nonclosed contours are then tracked twice, back and forth. The second possibility is to process open contours separately. In this case an additional code for the end of contour, or alternatively, information regarding the contour size, is needed. For a correct shape reconstruction from the contour information, the inside and outside of an object also need to be defined. A closed contour can be tracked and encoded either clockwise or counterclockwise. Depending of the convention used, one could then consider the left or right side of the oriented contour as the interior of an object.

Quite an extensive literature has already treated the problem of contour coding [75]–[80]. The most popular technique is chain coding, which consists in coding the position of the pixels in the contour relative to their neighbors. In the most classical chain coding, the first pixel in a contour is coded in its absolute coordinates. All the remaining pixels are then coded by just indicating their relative position compared to the last encoded neighboring pixel. Variants of this technique are used to improve the compression efficiency [68], [78]–[80].

Geometrical approximation methods constitute another family of contour-based representations that have been first investigated in the field of computer vision and pattern recognition. Geometrical representations include polygon approximation [81], [82] and cubic curves. Literature reports several schemes for the selection of the vertices representing the contour approximation for polygons [83], cubic curves [84], [85] or hybrid system [86], including both types for better shape representation.

Fig. 25 reports an example of different types of distortions that can be observed by using various lossy shape coding techniques.

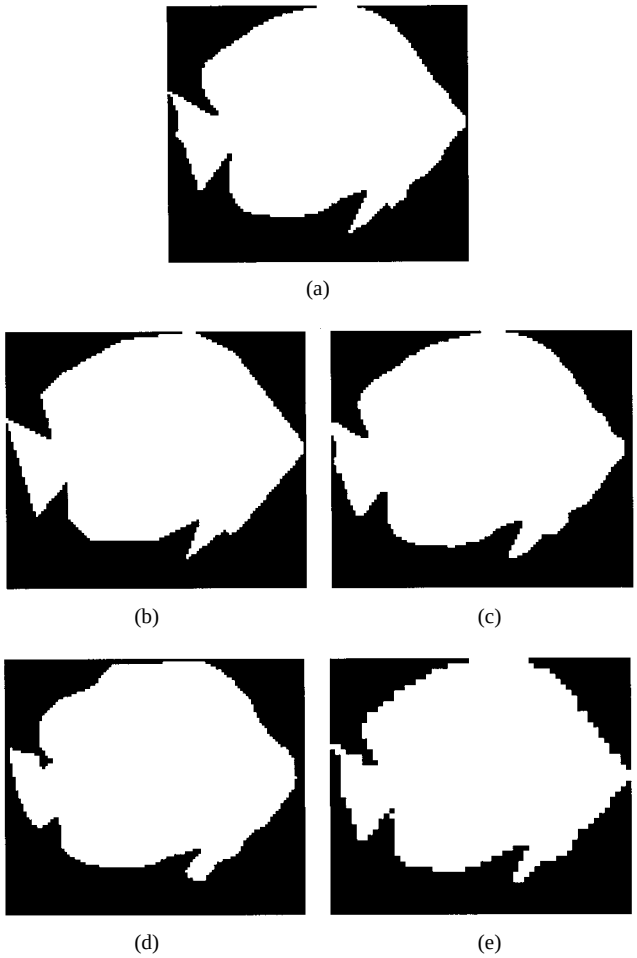


Fig. 25. Example of distortions observed using various lossy shape coding schemes: (a) original shape; (b) polygonal approximation; (c) lossy chain coding; (d) joint block/Hermite curve approximation; and (e) MPEG-4 VM block-based shape coding.

In a complete object-based compression scheme with lossy shape coding, the shape–texture coding interaction is of prime importance. The bit-rate reduction, resulting in a loss of shape coding quality, may actually increase the texture bit rate for a given texture quality. Preliminary studies [87], [88] have shown that the relation between the overall system R-D behavior and those of shape and texture are very complex. They depend on the image and also on coding parameters such as the desired quality, the bit rate, and the rate control scheme.

VI. QUANTIZATION

A. Introduction

To compress the data of the image, the subband signals are quantized. To perform an optimal quantization, the type of quantization (uniform, nonuniform) and the number of the quantization levels in the different subbands are defined.

B. Basic Equations of Quantization

Let X be a real random variable with probability density function (pdf) $p_X(\cdot)$. The cumulative distribution function

(cdf) $F(\cdot)$ is then

$$F(x) = \int_{-\infty}^{x} p_X(\xi)\, d\xi. \tag{39}$$

Let us give the following definition of a quantizer.

Definition 3: A device with input X and output $Y = \mathcal{Q}(X)$ is called a quantizer if $a_{i-1} < X \leq a_i$ implies $Y = \gamma_i$. The a_i's are called the quantization thresholds (decision levels) and the γ_i's are called the reconstruction levels [89].

The probability p_i that the output Y is γ_i depends on the cumulative distribution of X by

$$p_i = P(Y = \gamma_i) = F(a_i) - F(a_{i-1}). \tag{40}$$

Hence, the knowledge of $F(\cdot)$ gives us the possibility to compute the zeroth-order entropy rate R of the output

$$R = -\sum_i p_i \log_2 p_i \tag{41}$$

and the average rth order distortion D_r

$$D_r = E|Y - X|^r = \sum_i \int_{a_{i-1}}^{a_i} |x - \gamma_i|^r p_X(x)\, dx \tag{42}$$

where r is a positive integer. After quantization, a source coding algorithm is applied. The lower the entropy of the data, the more efficient the source coding algorithm (see Section VII). It can be concluded that in our context an optimal quantizer is one that minimizes D_r for a fixed entropy rate R.

C. Optimum Quantizers

It can be observed from (41) that the entropy rate R does not depend on the reconstruction levels γ_i. Therefore, the γ_i can be chosen to minimize the distortion D_r. It can be shown [89] that the optimal γ_i's are uniquely specified by a_{i-1}, a_i, and r by the equation

$$\int_{a_{i-1}}^{\gamma_i} |x - \gamma_i|^{r-1} p_X(x)\, dx = \int_{\gamma_i}^{a_i} |x - \gamma_i|^{r-1} p_X(x)\, dx. \tag{43}$$

For $r = 1$, (43) reduces to

$$\int_{a_{i-1}}^{\gamma_i} p_X(x)\, dx = \int_{\gamma_i}^{a_i} p_X(x)\, dx \tag{44}$$

or equivalently

$$E(X < \gamma_i | a_{i-1} < X \leq a_i) = \tfrac{1}{2}. \tag{45}$$

Equation (45) defines the median point of the interval. We can also compute (43) for $r = 2$

$$\gamma_i = \frac{\int_{a_{i-1}}^{a_i} x p_X(x)\, dx}{\int_{a_{i-1}}^{a_i} p_X(x)\, dx} \tag{46}$$

which is the same as

$$\gamma_i = E(X | a_{i-1} < X \leq a_i). \tag{47}$$

Equation (47) defines the expected value of the interval which is in general not the same as the median point.

The design of a quantizer in the minimum mean square error (MMSE) sense reduces to choosing the thresholds a_i to minimize D_2 by respecting (47). This yields a set of nonlinear equations which are difficult to solve in general.

D. Uniform Quantizer

The simplest quantizer is the uniform quantizer which is defined as follows.

Definition 4: A quantizer is uniform when the quantization thresholds are uniformly separated by the same distance, hence

$$a_i - a_{i-1} = \Delta, \quad \forall i. \tag{48}$$

A very surprising fact is that this simple quantizer is optimal or quasi-optimum in most of the cases. Actually, it is optimal if the input X has a Laplacian or exponential pdf. Otherwise, the optimal quantizers perform only negligibly [89] better than the uniform quantizer. This is the reason for the popularity of uniform quantizers.

E. Successive-Approximation Quantization (SAQ)

1) Introduction: As shown in Section VI-D, the uniform quantizer is well suited in an image-compression framework; however, it does not permit the definition of a progressive transmission and an exact rate control. To improve the image-compression scheme featuring these new functionalities, different quantization strategies have to be adopted. An approach achieving this goal is SAQ of the coefficients. It is detailed below.

2) Classical SAQ: To exploit the existing zero-correlation across subbands, the EZW algorithm—proposed by Shapiro [43]—is based on three main blocks: 1) the hierarchical subband decomposition; 2) the prediction of the absence of significant information across scales using zerotrees; and 3) the entropy-coded SAQ.

The first block is a subband decomposition of any type. All decompositions described in Section IV-B can be used for that purpose.

The second block is the heart of the EZW algorithm and goes as follows. For each subband, a parent–children relationship is defined. The low-frequency subband is defined to be the parent of all other subbands. Then, a parent–children relationship is defined for all subbands. An example of a parent–children relationship for the wavelet decomposition is shown in Fig. 26.

The next block is the scan of all coefficients to detect the highest valued coefficient (in magnitude) $c_{\max}$ of all the subbands. This coefficient is used to compute an initial threshold T_0 by the following equation:

$$T_0 = \gamma \cdot c_{\max} \tag{49}$$

where γ is a parameter of the system. Convergence of the quantization is ensured if $\gamma \geq \frac{1}{2}$. Then different scans are performed, producing a stream of symbols.

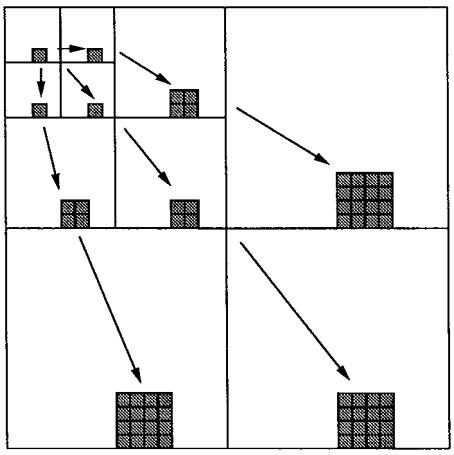


Fig. 26. Parent–children relationship for the wavelet decomposition of the EZW algorithm.

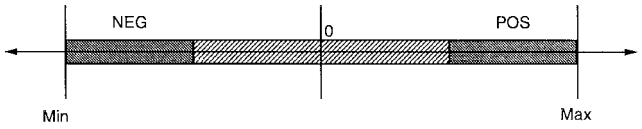


Fig. 27. Illustration of the primary scan of the zerotree wavelet algorithm. The unquantized coefficients range from $\min$ to $\max$ and are quantized to NEG, POS, or zero.

The last block is the arithmetic coder, which turns the symbols into the final bit stream.

The successive approximation consists in an iteration over two scans, called the primary scan and the secondary scan. The primary scan locates the significant wavelet coefficients, and the secondary scan refines the values of all considered coefficients so far.

The symbol stream of the primary scan is generated for the coefficients in the subbands using the following symbols: POS (positive) if the coefficient is significant with respect to the threshold T_i and positive; NEG (negative) if the coefficient is significant with respect to the threshold T_i and negative; ZTR (zerotree) if the wavelet coefficient is insignificant with respect to the threshold and all its children are insignificant, too; IZ (isolated zero) for all other wavelet coefficients.

This quantization procedure is illustrated in Fig. 27. One can see that the uncertainty interval $\mathcal{C}$ may be not uniform. Indeed, with $\gamma \geq \frac{1}{2}$, the uncertainty interval of the zero-quantized coefficients is always larger than for the values quantized to POS or NEG. In the special case of $\gamma = \frac{1}{2}$, this dead zone is twice as large as the uncertainty interval of the significant coefficients.

In the secondary scan, all the coefficients which have not been quantized to zero so far are refined. This is done by transmitting, for each of these coefficients, a binary value telling if the unquantized coefficient is in the lower or upper half of the interval. This refinement is illustrated in Fig. 28 for the case where the coefficient was primarily scanned as POS.

After a secondary scan, a new threshold is computed as

$$T_{i+1} = T_i/2 \tag{50}$$

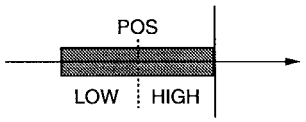


Fig. 28. Illustration of the secondary scan of the zerotree wavelet algorithm. Each coefficient which has been quantized to a significant value in the previous primary scan is refined by a factor of $\gamma = 2$.

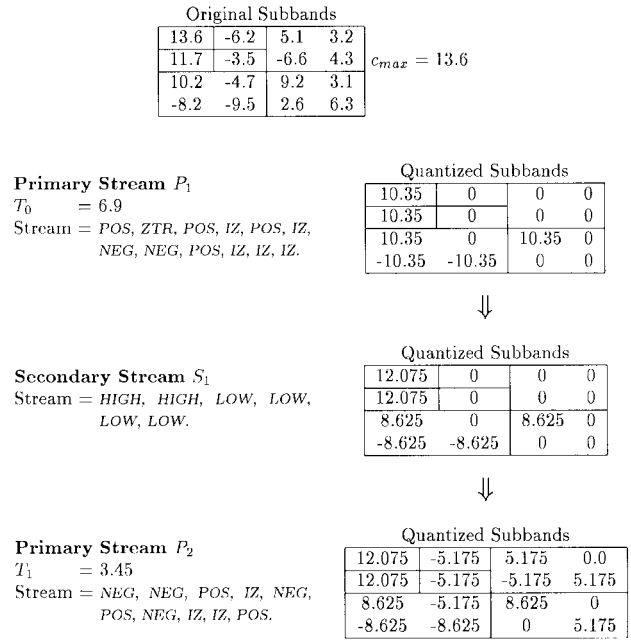


Fig. 29. Example of a classical SAQ on wavelet coefficients. Illustrated are the first two primary streams (P_1 and P_2) and the first secondary stream (S_1). The parent–children relationship is that of Fig. 26, and $\gamma = 2$.

where i is the iteration number. The algorithm is iterated by applying successive primary and secondary symbol streams. These are then encoded using the entropy coder. The result is a completely embedded bit stream. An example of such a quantization is shown in Fig. 29. The EZW algorithm has shown excellent performances in compressing natural images.

One approach for generalizing the EZW algorithm is as follows. At each iteration, instead of dividing the threshold by two, let us define the new threshold by the general formula

$$T_{i+1} = T_i/Q \tag{51}$$

where Q is a floating point value. Moreover, instead of splitting the uncertainty interval of the nonzero coefficients into two bins, a refinement into M distinct values is produced in the secondary stream. If $Q = 3.0$ and $M = 3$, one approaches a uniform quantization very closely. This is illustrated in Fig. 30.

One can observe that after the first primary stream, a uniform quantization is achieved if Q is chosen to be 3.0. Closer examination shows that this is indeed the case after each primary stream. The EZW algorithm proposed in [43]

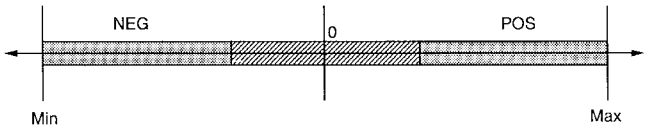


Fig. 30. Illustration of the primary scan of the zerotree wavelet algorithm with $Q = 3.0$. The unquantized coefficients range from $\min$ to $\max$ and are quantized to NEG, POS, or zero. The three uncertainty intervals are equally large.

has the following system parameters: $\gamma = 0.5; Q = 2.0;$ and $M = 2$.

3) SAWT and SAND SAQ: To use the described shape-adaptive transforms in an efficient coding environment, it is proposed to generalize the EZW to match these transforms. Special care has been dedicated to retain all the advantages of the EZW algorithm for arbitrarily shaped regions. As for the original EZW algorithm, the proposed technique is based on three basic blocks, namely the transform (SAWT or SAND), the zerotree prediction, and the SAQ.

After performing the subband transform, the zerotree prediction in turn allows for a further improvement of the energy compaction by taking into account the remaining interband correlations. The zerotree prediction has to be generalized to match the properties of arbitrarily shaped subbands. Indeed, in the case of rectangular images the subbands are rectangular as well. This allows a straightforward definition of the parent–children relationship as described in Section VI-E.

In the case of arbitrarily shaped regions, it may happen that a group of children has no parent or that a parent has no children. This leads to a redefinition of the parent–children relationship for arbitrary shapes. The idea behind the zerotree is to be able to code a complete tree of zero-valued coefficients with only one symbol. Hence, it is important that a missing pixel in a whole tree does not affect the efficiency of the zerotree symbol. The proposed approach is defined as follows:

1) create a zerotree in the same way as for rectangular images;
2) consider all the pixels of the tree not belonging to the arbitrarily shaped regions as quantized to zero;
3) code the tree as before, but considering only pixels in the shape.

This approach ensures that the maximum number of zerotrees can be found. The method is illustrated in Fig. 31. Fig. 31(a) shows the complete zerotree where the black pixels belong to the region and the gray pixels do not. Fig. 31(b) is the purged tree. Note that the appropriate placing of the pixel in the subbands, as described in Section V-D, allows for an efficient representation with zerotrees. An example of the SAQ encoding for arbitrarily shaped regions is shown in Fig. 32.

Finally, the SAQ provides an embedded bit stream and allows for an exact rate control. The embedded bit stream together with the multiresolution structure allows for additional features such as progressive transmission, retrieval, and data browsing for each region independently.

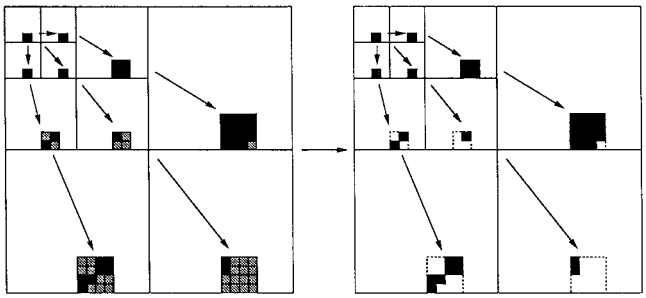


Fig. 31. Illustration of the generalized zerotree prediction. (a) The complete zerotree. Black pixels belong to the region. Gray pixels do not belong to the region. (b) The purged tree.

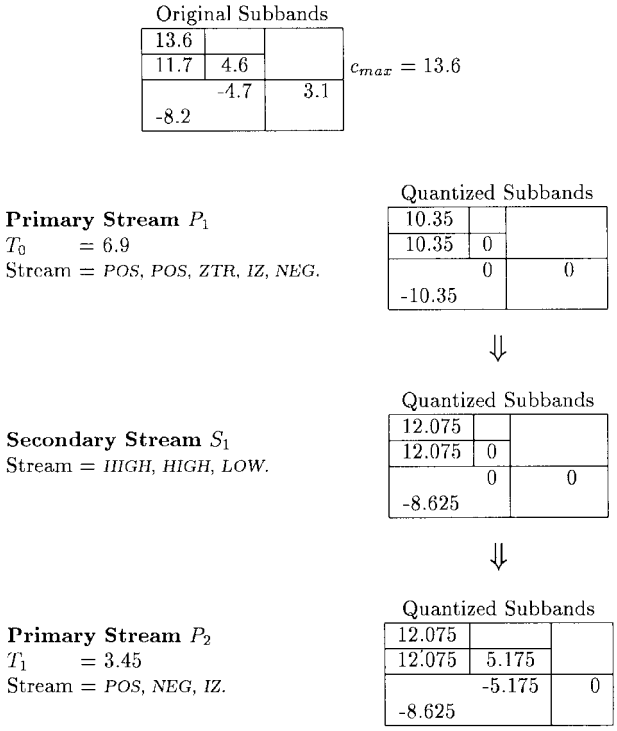


Fig. 32. Example of an SAQ on wavelet coefficients of an arbitrarily shaped region. Illustrated are the first two primary streams (P_1 and P_2) and the first secondary stream (S_1). The parent–children relationship is that of Fig. 26.

4) Lossless SAQ: The two conditions for lossless compression in a multiresolution environment have been stated in Section IV-D. It is clear that even though these conditions on the filter bank might be satisfied, they are not sufficient to render the entire scheme lossless. To render this compression algorithm lossless, the following four conditions on the quantization are considered.

1) The initial threshold should be set to $T_0 = M^n$ with n being an integer value.
2) The threshold is divided by M at each iteration, $T_{i+1} = T_i/M$.
3) The iterative quantization procedure is iterated n times for the primary stream and $n - 1$ times for the secondary stream.

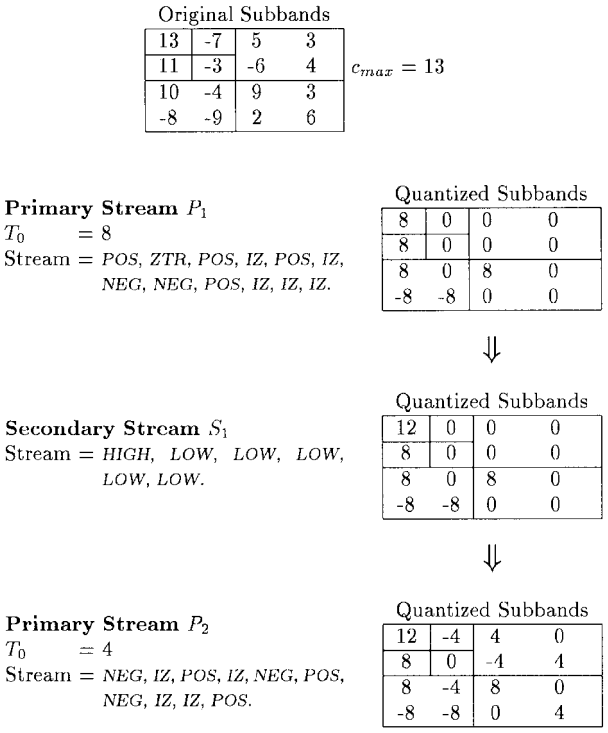


Fig. 33. Example of a lossless SAQ on wavelet coefficients. Illustrated are the first two primary streams (P_1 and P_2) and the first secondary stream (S_1). The parent–children relationship is that of Fig. 26.

4) The reconstruction levels have to be set to the minimum absolute value of the uncertainty interval of the reconstruction.

There are two ways to achieve lossless reconstruction. The first way is if, at the last primary scan, the uncertainty interval $\mathcal{C}$ for the reconstruction is $\mathcal{C} \leq 1$, and if the values to be reconstructed are integers. This can be reached if conditions 1)–3) are met. This ensures that after the nth iteration of the primary stream, the threshold will be exactly one. At the nth primary stream, for each quantized coefficient c, the following inequality will be satisfied:

$$c_{\min} \leq c < c_{\max} \tag{52}$$

where $c_{\min}$ and $c_{\max}$ are the bounds of $\mathcal{C}$. After n iterations of the primary stream $c_{\max} - c_{\min} = 1$. Since c is a signed integer value, the correct value of c must be

$$c = c_{\min}. \tag{53}$$

The second way lossless reconstruction is achieved is when conditions 1) and 2) are not satisfied, but enough iterations (normally more than n) are considered. The stop criterion is again $\mathcal{C} \leq 1$, allowing for a perfect mapping of the quantized coefficients to their integer values.

An example of lossless quantization with $M = 2$ is shown in Fig. 33.

F. Vector Quantization (VQ)

In contrast to scalar quantization discussed in previous sections, VQ is the process of mapping sets of values in form of vectors into a predefined set of patterns. Although VQ can be seen as a generalization of scalar quantization, it can be shown that from an R-D point of view that it results in an optimum performance even when the data to be quantized are made of independent samples [90]. Let us give the following definition of a vector quantizer.

Definition 5: A device with input $X = (x_1, x_2, \ldots, x_k)$ and output $Y = \mathcal{Q}(X) = (y_1, y_2, \ldots, y_k)$ is called a vector quantizer $\mathcal{Q}$ of dimension k and size N if any vector X is mapped into one of the N possible output points Y_i contained in the finite set C defined as

$$C = \{Y_i | Y_i \in \mathcal{R}^k, i \in \mathcal{I}\} \quad \text{with} \quad \mathcal{I} = \{1, 2, \ldots, N\}. \tag{54}$$

The set C is called the codebook and the output or reproduction points Y_i are called code vectors. The resolution or rate of the vector quantizer is defined as $r = (\log_2 N)/k$. A partition of $\mathcal{R}^k$ into N regions R_i, called cells, is associated with the codebook and is defined as

$$R_i = \left\{X | X \in \mathcal{R}^k, \mathcal{Q}(X) = Y_i\right\} \tag{55}$$

with

$$\bigcup_i R_i = \mathcal{R}^k$$

and

$$R_i \cap R_j = \emptyset \quad \text{for } i, j \in \mathcal{I}, i \neq j.$$

Taking into account the above definitions, VQ can be interpreted as an association of every code vector Y_i to a cell R_i, in which the actual quantization process would be the assignment of Y_i to X when $X \in R_i$.

In principle, the design of a VQ requires a common definition of codebook used in both encoder and decoder. Encoder and decoder then only exchange the indexes $i \in \mathcal{I}$ of the Y_i instead of the values of X.

The choice of the codebook depends on the statistical properties of the source data to be quantized and as a consequence plays an essential role in the performance of VQ. Several strategies have been developed in order to design appropriate codebooks in an efficient way requiring minimal computational load during quantization. The most well-known approach for the design of a codebook is that of generalized Lloyd algorithm, also known as LBG algorithm [91]–[93]. In the LBG approach, the codebook is designed using a training set containing data of the same kind as that to be quantized. The codebook is then stored and used in both encoder and decoder. Variants of VQ may update the initial codebook as the process of quantization is in progress. Such quantizers are known as adaptive vector quantizers.

Another class of vector quantizers makes use of no explicit codebook in order to avoid storage requirements. Lattice VQ is one such approach. As for scalars, the optimal high-resolution entropy-constrained scalar quantization is the uniform quantization (see Section VI-D); similarly, for large vector sizes, the optimal vector quantizer for a

constrained entropy is obtained in using a codebook with uniform cells [95].

Formation of input vectors is another important issue in the design of vector quantizers applied to image coding. In earlier realizations, vectors were formed by pixel values in a square block of an image [92]. In more recent techniques, vectors are created from the coefficients of a transformed image. Either coefficients of a given subband around the same spatial position or coefficients from different subbands in the same spatial position can be used. Some variants use both kinds of coefficients to form vectors [95].

Although from a theoretical point of view it is possible to show that VQ remains superior to scalar quantization, the added complexity does not pay off for most applications.

G. Trellis Coded Quantization (TCQ)

TCQ is a source coding method offering similar R-D performance to that of VQ at only slightly higher complexity than scalar quantization. The principle behind TCQ is motivated by alphabet-constrained R-D theory, which determines the best achievable performance when compressing data using a finite set of quantized symbols. Notions used in TCQ are similar to that of trellis coded modulation (TCM) widely used today for the design of voice line modems [96]. In its simple form, TCQ exploits a codebook of 2^{B+1} codewords when coding a given source at a bit rate of B bits/symbol [97]. This is twice the number of codewords needed in a conventional quantization mechanism. It is possible to show that this doubling of number of codewords allows one to obtain nearly all the theoretical gain possible over a B bits/symol Lloyd–Max quantizers. The codebook is partitioned into 2^{P+1} subsets with $P \leq B$ using the set partitioning mechanism proposed by Ungerboecks in TCM. Typically, the expanded codebook is divided into four subsets $(P = 1)$, each containing 2^{B-1} codewords such that the minimum distance between the codewords in a subset is achieved. The process of quantization then proceeds as follows. To encode a sequence of N symbols $x = \{x_0, \ldots, x_{N-1}\}$ at R bits/symbol, N stages of a trellis are cascaded together. Fig. 34 gives an example of one stage of such a trellis with eight states with codebook partitioning into four subsets, D_0, D_1, D_2, and D_3. In this example, four scalar quantizations are performed at each stage (one for each partition) to determine the best quantization level (in general based on the minimum mean-squared criteria). The quantization error introduced by this process is chosen as the cost function when selecting the path in the trellis from one stage to the next. The Viterbi algorithm can be used to find the trellis path with the minimum cumulative distortion between the source and its quantized version [98]. The sequence of resulting quantization codewords can be identified using R bits at each stage of the trellis, namely, one bit to specify the path in the trellis, and $R-1$ bits to select the desired quantization level in the partition associated to that path. Variants of TCQ have been reported in the literature for both scalar and VQ [99]–[101]. More recently, a technique using wavelet decomposition and TCQ was reported in [102] with superior results for progressive compression of still images. Results are shown in Section X.

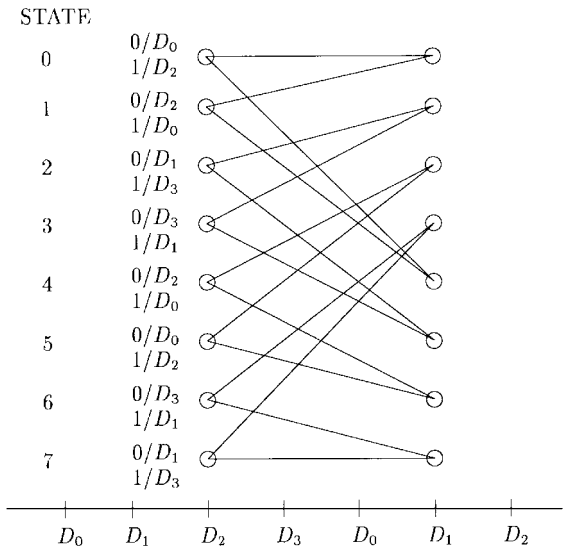


Fig. 34. Transitions in a stage of a trellis coder.

Table 1
Sample Set of Symbols to be Represented in Binary Code

DCT Coefficient value	0	10	20	40	60	120
Frequency of occurrence	29	26	23	19	2	1
Fixed-length code	000	001	010	011	100	101
Variable-length code	11	10	01	001	0001	0000

VII. Codeword Assignment

A. Variable Length Coding

The message extraction phase described in the previous sections results in a list of symbols. Codeword assignment is the procedure of mapping the symbols to bit patterns, suitable for transmission or storage. To illustrate the different schemes presented in this section, we will refer to the following example.

Suppose we get a collection of 100 quantized DCT coefficients as our list of symbols. Let us assume only the values shown in Table 1 occur in this collection, with the indicated number of occurrences. The bit patterns assigned to each symbol of the set are called the bit codes. A straightforward conversion is to assign a fixed-length code to each symbol. The length of the code depends on the number of different values which appear in the set, also referred to as the alphabet. For an alphabet of size S, the length of the code is $\lceil (\log_2 (S) \rceil$. In the example of Table 1 we have six values leading to a code length of three. All symbols have to be defined as unique. An example fixed length bit code is given in Table 1. Encoding our 100 symbols with this code results in 300 bits.

At this point of the compression of the image, it is very likely that the number of symbols that represents the input picture is the same as the original number of values describing the pixels values. In case of segmentation-based schemes, it might even be higher because of the

description of the shape itself. The message extraction is a map for input signal (pixels) onto a set of symbols presenting statistics which the code assignment can exploit.

With the symbols statistics, e.g., the probability of occurrence $p(s_i)$ of each symbol s_i, we could find a bit code which is more efficient than the fixed-length code from above [103]. The variable-length code shown in Table 1 represents the same set of 100 symbols on only 225 bits, which is a gain of 25% over the previous scheme.

The measure of the goodness of a set of bit-codes is the zeroth-order entropy, defined as

$$H_s = -\sum_{i=1}^{S} p(s_i) \log_2 \left(p(s_i)\right) \tag{56}$$

where $\log_2 [p(s_i)]$ represents the actual number of bits necessary to encode the symbol s_i and $p(s_i)$ its probability of occurrence. The above definition gives the theoretical lower bound where we get the optimal length for each of the codes. When evaluating a bit code, one replaces this length with the actual bit length of the code for all the s_i. It can be shown that this function is minimum when the code is optimum [104], and therefore enables the objective comparison of two bit codes.

1) Adaptive Schemes: To retrieve the symbols from the bit sequence, the decoder has to know the exact bit code which has been used. For an efficient representation, the bit code has been built on the statistics of the symbols. Therefore, the decoder could also rebuild the same bit code providing it gets the statistics. Some image-compression standards simply use a predefined bit code based on statistics from a set of typical input images. As those might change from image to image, or might change for different parts of an image, the overhead of encoding the statistics along with the symbols might well be compensated with the better compactness of the bits representing the symbols. One way to embed the statistical information is to start with a predefined statistic, update it at each encoded symbol, and use the new statistical information for the next symbol. The decoder can then do the same procedure, updating the statistical information at each decoded symbol. Such a scheme is called adaptive. The overhead of transmitting the statistical distribution of the symbols is spread over each of them, as only the last one is coded with the bit code most suited to the true statistics of the symbols. How to update the probabilities of occurrence of each symbol can be found in [105] and [106].

B. Runlength Coding

A first approach to reduce the amount of data is to describe the sequence of symbols in term of bursts of the same symbol (run) and the number of times this symbol is found in a row (length), hence the name. A simple example is given in row *A* of Table 2. Note that the runs consist of symbols from the original alphabet, whereas the lengths are new symbols. Compression can be gained in assigning a fixed-length code to each of the runs and lengths to be encoded. Different codes have to be used for runs and lengths, as the size of the original alphabet is independent from the maximum burst length.

Table 2
Runlength Representation of a Sequence of Symbols

Symbols sequence		10	0	0	0	0	10	10	20	10	0	0	0
A	Runs	10				0		10	20	10			0
	Length	1				4		2	1	1			3
B	Coefficient	10					10	10	20	10			
	Zeros Length	4					0	0	0	3			

In real image compression algorithms [107]–[109], however, the runs and lengths are encoded into bits with a variable-length scheme like a Huffman or arithmetic coding. Two codes are used for runs and lengths, taking into account their different statistical distributions. Furthermore, the representation might slightly change if more precise information is available about the original symbol statistics. The quantization and ordering of the DCT coefficients in [107] and [108] generally lead to long sequences of zeros, interleaved with a few nonzero coefficients. Therefore, the two-symbol representation is of type (nonzero coefficient; zero-sequence length). The corresponding symbols are shown in row *B* of Table 2.

Runlength is not a proper coding scheme as it only produces another set of symbols, thus making it simply a representation change. The type of statistical feature one can exploit is that the input data consist of long bursts of the same symbol value. In the worst case where any two successive symbol values are different, the number of symbols after the runlength is doubled.

C. Huffman Coding

The goal here is to give longer codes to less probable symbols. Given an alphabet of size S with symbols s_i and their probability of occurrence $p(s_i)$, Huffman's scheme [110] of building the bit codes c_i tries to minimize their average length

$$E[\text{length}(c_i)] = \sum_{i=1}^{S} p(s_i)\text{length}(c_i). \tag{57}$$

For the encoded sequence of symbols to be uniquely decodable, the bit codes c_i have to form a prefix code, i.e., no c_i is a prefix of another one. In this case, at any bit-code boundary in the encoded sequence of symbols only one symbol will match, and the corresponding bits can be removed from the bit stream to go to the next bit-code boundary.

The bit-code construction discovered by Huffman [110] satisfies the prefix-code constraint while leading to an optimal code. The expected bit-length length(c_i) comes quite close to the lower bound given by the entropy of the symbol sequence given in (56).

Fig. 35 illustrates how a Huffman code can be built. The symbols and their probabilities have been taken from Table 1. The codes with the highest occurrence frequency are expected to get shorter bit codes. As we are building a prefix code, the length of the two longest codes must be

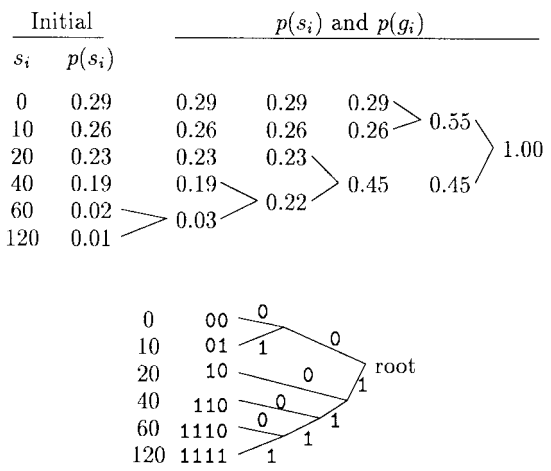


Fig. 35. Huffman bit-code construction example.

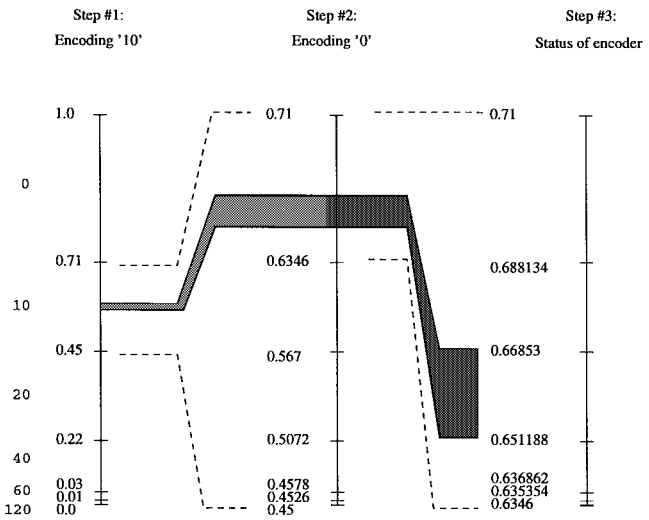


Fig. 36. Example of coding three symbols with an arithmetic coder.

equal. Otherwise, in removing the last bit of the longest we would get another valid prefix code but with a shorter length, rendering the first one nonoptimal. In general, one can build a bit code in which the two longest codes differ only in the last bit.

To get the bit codes for all the symbols, one can build a tree in combining the two symbols with the lowest $p(s_i)$ into a single group symbol g_i with frequency of occurrence equal to the sum of the two $p(s_i)$. Doing this recursively on the obtained s_i and g_i leads to a binary tree, on which we can put bit labels to each branch, as shown in the lower part of Fig. 35. The bit code for a symbol of the alphabet is then given by the bit label sequence of the branches when going from the root to the leaf of the binary tree. The bit codes for this example are shown in the last row of Table 1. As mentioned before, this optimal Huffman code leads to a 25% reduction of the bits necessary to encode the symbol sequence.

From this construction, it is clear that Huffman bit codes have a minimum bit length of 1 bit per symbol. This also gives a lower bound on the length of the produced bit stream. In applications where symbol frequency of occurrence may be far from uniformly distributed, this is considered a drawback. Having an integer number of bits per symbol helps synchronizing encoder and decoder and results in a less error-prone system. However, having some synchronization point on bit boundaries can be used to detect the end of the list of encoded symbols, as "no bits left" means "no symbols left."

Bit boundary is also helpful in case of bit errors. For this purpose, reversible Huffman codes are used, which enable one to decode from the end of the bit stream. Therefore, reversible Huffman codes are prefix and postfix codes, which is not guaranteed in the above mentioned construction of the bit codes (e.g., in Fig. 35 the code for symbol 40 is a postfix of the code for symbol 60). The group of symbols before the first bit error and the group after the last bit error can therefore be decoded. For single bit errors, one can come up with a scheme decoding all but the corrupted symbols from the bit stream. This method is used in the decoding in [111].

D. Arithmetic Coding

The main idea behind arithmetic coding is the encoding of a number of arbitrary precision describing the sequence of symbols of length N. Any other sequence $\{s_i\}$ of length N would lead to a different number. The precision of the number will be the minimum to achieve this distinction between sequences of the same lengths, and its binary representation is the actual encoding of the string of symbols.

The construction and encoding of this number can be done incrementally. Fig. 36 shows three steps of construction of the number comprised between zero and one. This range is divided into slots whose size is proportional to $p(s_i)$. Therefore, numbers whose values lie in the slot for symbol 0 represents sequences starting with this symbol. For a sequence of a single symbol 0, any number in this range will describe the sequence.

For further symbols, the range corresponding to the encoded symbol is again subdivided into slots corresponding to the (possibly updated) $p(s_i)$. The same process restarts, until all symbols have been encoded and the range of values for the number representing this sequences is known. In the example of Fig. 36, the sequence {10} leads to the number range [0.45, 0.71], the sequence {10, 0} to the range [0.6346, 0.71], and the sequence {10, 0, 20} to the range [0.651 188, 0.668 53].

Arithmetic coding has the interesting feature of being able to get below the 1-bit-per-symbol limit. It therefore achieves better compression, especially for longer sequences of symbols as their true frequency of occurrence is far better represented in the model. A drawback of this scheme, however, is that the number of symbols cannot be embedded in the bit stream and therefore has to be coded separately. Furthermore, the same number represents sequences of any length sharing the N first symbols. Arithmetic coding is also much more sensible to bit errors, as modifying a single bit might alter a great number of symbols. Furthermore, backward decoding is impossible.

The implementation of an arithmetic coder leads one to consider numbers of arbitrary precision. Those practical issues, as well as an efficient solution, are given in [112] and discussed further in [113].

VIII. Rate Allocation

A. Introduction

Image transformations or decompositions for coding purposes have the goal to compact most of the energy of the image in as few pixels as possible. For example, the purpose of performing the DCT of image blocks is that most of the information will be around DC, and hence many transform coefficients corresponding to higher frequencies will have values near zero. In SBC of images, the image is split into frequency bands, and again one band will contain most of the information, whereas other bands will have very little energy. It is clear that the better the energy compaction the better the coding performance will be. This criterion can be defined in a precise way and is called unified coding gain, which is the subject of Section VIII-B. Having split the image into several subbands, another problem consists of optimally assigning quantizers to these subbands, which are to be coded at a given bit rate. This problem is called bit allocation.

B. Unified Coding Gain

Coding gain is a performance measure coming from R-D theory and is described in detail in [114]. This criterion denoted G_{TC} measures the performance of a block transform. It is defined in [42]; formally

$$G_{TC} = \frac{\frac{1}{M}\sum_{k=0}^{M-1}\sigma_k^2}{\left[\prod_{k=0}^{M-1}\sigma_k^2\right]^{1/M}} \tag{58}$$

where σ_k^2 are transform coefficient variances and M is the number of coefficients. This measure assumes that all coefficients, as well as the original signal, have the same type pdf [114]. This assumption is strictly correct only for Gaussian sources. In the context where the sources are natural images, it is not exactly satisfied. Nevertheless, it is known from literature that this measure is consistent with the observed experimental coding performance for block transforms. The higher the gain G_{TC}, the better the performance of the transform.

1) Coding Gain of Orthonormal Filter Banks: In the special case of orthonormal filter banks such as QMF's or CQF's, the coding gain has been introduced by Akansu *et al*. [115] and is given by

$$G_{\text{ORTH}} = \frac{\frac{1}{M}\sum_{l=0}^{M-1}\sigma_l^2}{\left[\prod_{l=0}^{M-1}\sigma_l^2\right]^{1/M}} \tag{59}$$

where σ_l^2 is the variance of the signal in the lth subband, and M is the number of subbands. This formula holds for a uniform decomposition implying equal bandwidths. It should be emphasized that this measure is only valid for orthonormal filter banks. This criterion can be included in the design of filter banks and has been done [29], [116] with some success.

It is known that the restriction of the filter bank being orthonormal is too strong. Good subband filters [44], [117] are not included in this class. The generalization of the criterion to arbitrary subband-like decompositions has been published in [118] and is described in the following section.

2) Unified Coding Gain—General Case: Let us define the parameters A_i and B_i by the following equations:

$$\sigma_i^2 = A_i\sigma_{in}^2 \tag{60}$$

and

$$\sigma_r^2 = \sum_{i=0}^{M-1} B_i\sigma_{q_i}^2 \tag{61}$$

where σ_i^2 is the variance of the ith subband, σ_{in}^2 is the variance of the input signal, σ_r^2 is the variance of the reconstructed image, $\sigma_{q_i}^2$ is the variance of the quantization error of the ith subband, and M is the number of subbands. Also, define the inverses of the downsampling factors by α_i, where

$$\sum_{i=0}^{M-1}\alpha_i = 1. \tag{62}$$

Under the assumption of uncorrelated quantization errors, B_i is given by

$$B_i = \alpha_i\sum_{n=0}^{L_i-1} g_i^2[n] \tag{63}$$

with L_i the number of coefficients of the ith synthesis filter.

The unified coding gain [118] is then given by

$$G_{\text{SBC}} = \frac{1}{\prod_{i=0}^{M-1}\left(\frac{A_iB_i}{\alpha_i}\right)^{\alpha_i}} \tag{64}$$

which can be expressed as

$$G_{\text{SBC}} = \frac{\sigma_{in}^2}{\prod_{i=0}^{M-1}\left(\sigma_i^2\sum_{n=0}^{L_i-1} g_i^2[n]\right)^{\alpha_i}}. \tag{65}$$

Note that (65) simplifies to

$$G_{\text{SBC}} = \frac{\sigma_{in}^2}{\left(\sum_{n=0}^{L-1} g^2[n]\right)\cdot\left(\prod_{i}^{M-1}\sigma_i^2\right)^{1/M}} \tag{66}$$

for orthogonal filter banks and equal downsampling factors. Equation (66) is similar to (59).

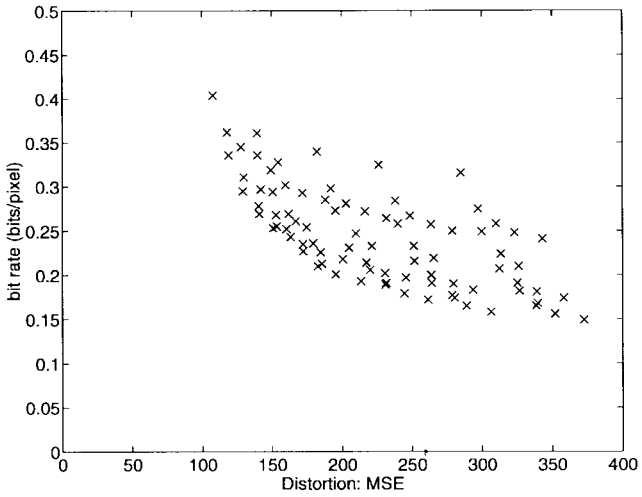


Fig. 37. R-D plot of all possible bit allocations in the case of N subbands and M admissible quantizers per subband.

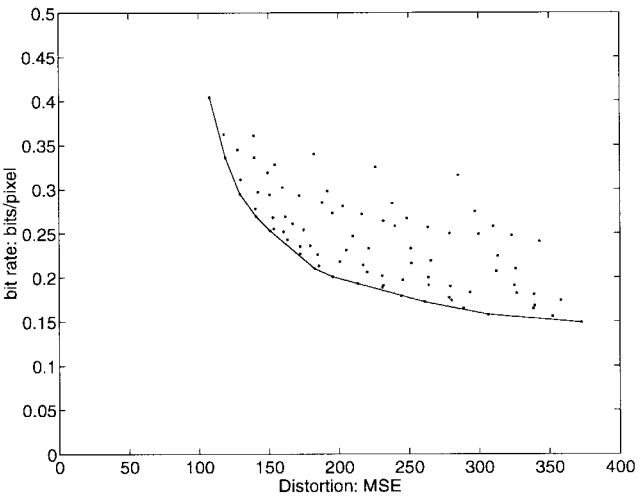


Fig. 38. The convex hull of a R-D plot.

C. Optimal Bit Allocation

The solution to optimal bit allocation was first proposed by Westerink *et al.* [119]. Simplifications of that algorithm have then been proposed in [120] and [121]. In bit allocation, as used in SBC, the objective is to optimally assign to each subband its own quantizers, such that the total number of bits equals a given quota and the overall distortion is minimal.

In subband-like coding schemes, the bit allocation problem can be formulated as the problem to minimize the overall quantization error D

$$D = \sum_{n=1}^{M} d_n \tag{67}$$

under the constraint of a given total bit rate R

$$R = \sum_{n=1}^{M} r_n \tag{68}$$

by assigning to each subband the appropriate quantizer having a distortion d_n and a rate r_n. It has to be emphasized here that the only assumption made about the overall distortion is that it can be written as a sum of the individual distortions. However, no assumption about the nature of the distortion measure is made. Moreover, the quantizers do not necessarily have to be of the same nature either; each subband can have its own distortion measure and its own admissible quantizers.

The MSE will be used as a measure of the distortion, although it is clear that it is not a good measure at all for the quality of natural images when the human visual system is to be evaluated.

1) R-D: In order to explain the problem, consider L subbands, each of them having Q admissible quantizers. Let us imagine that for all Q^L different realizations, the rate and the distortion are computed. The output of such a simulation is illustrated in Fig. 37 in the case of subband decomposition of the test image "Pepper."

The optimal points are those lying on the lower convex hull on the plot of all possible bit allocations [122]. The convex hull of the previous example is shown in Fig. 38. Finding a convex hull of a set of points has been investigated in literature, and many fast algorithms have been designed to solve the problem [123]. However, exhaustive computation of all the combinations is not practical. The desired algorithm should find the convex hull in a limited number of computations. Such an algorithm has been derived by Westerink *et al.* in [119].

Each bit allocation in the R-D plot is the result of assigning a certain quantizer to each subband. Let us denote the assignment of quantizer v to subband l by v_l. Using this notation, each possible bit allocation is described by the vector

$$\underline{v} = (v_1, \ldots, v_L). \tag{69}$$

Equations (67) and (68) can then be rewritten in the following way:

$$D(\underline{v}) = \sum_{n=1}^{N} d_l(v_l) \tag{70}$$

$$R(\underline{v}) = \sum_{n=1}^{N} r_l(v_l). \tag{71}$$

A useful property satisfied by all the points on the convex hull is that if we draw a line through neighboring points on the convex hull, then all other points of the set must lie on or at one side of this line. This is illustrated in Fig. 39. Denote two points of the convex hull by the vectors $\underline{l}$ and $\underline{k}$. Moreover, define the line through these points by $L(\underline{l}, \underline{k})$. This allows to define a convex hull by using the equation of the line $L(\underline{l}, \underline{k})$ [119]. Thus, the convex hull is defined as the set of points satisfying the following equation:

$$[R(\underline{v}) - R(\underline{k})] \geq S(\underline{l}, \underline{k}) \cdot [D(\underline{v}) - D(\underline{k})] \quad \forall \underline{v} \tag{72}$$

where $S(\underline{l}, \underline{k})$ is the slope of the line $L(\underline{l}, \underline{k})$, which can be computed by

$$S(\underline{l}, \underline{k}) = R(\underline{l}) - R(\underline{k}) D(\underline{l}) - D(\underline{k}). \tag{73}$$

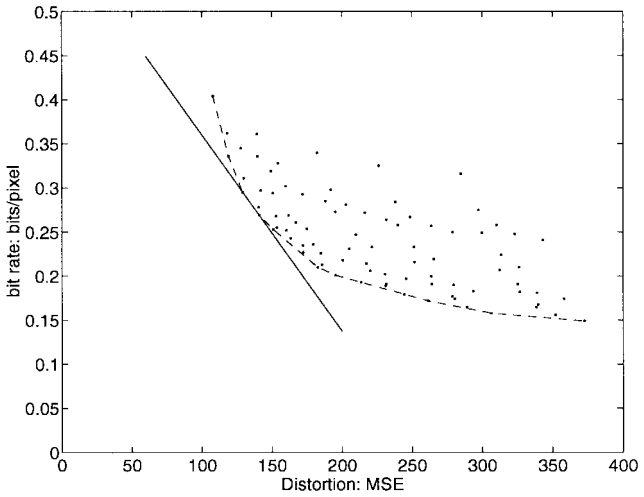


Fig. 39. A line through two neighboring points on the convex hull.

Now, by substituting (70) and (71) in (72) we obtain

$$\sum_{n=1}^{N}[r_n(v_n)-r_n(k_n)] \geq S(\underline{l},\underline{k})\sum_{n=1}^{N}[d_n(v_n)-d_n(k_n)]. \tag{74}$$

The term $S(\underline{l},\underline{k})$ does not depend on the running variable n and can be put into the sum. After having moved everything to the left-hand side, the inequality becomes

$$\sum_{n=1}^{N}\{[r_n(v_n)-r_n(k_n)]-S(\underline{l},\underline{k})[d_n(v_n)-d_n(k_n)]\} \geq 0. \tag{75}$$

This inequality is the basis of the optimal bit allocation algorithm. It gives the relationship of two neighboring points of the convex hull. Let us assume we know one point of the convex hull, then the neighboring point can be found by (75). Actually, there will be two solutions since a point always has two neighbors. However, assume we begin at maximal distortion and minimal rate, this point will always lie on the convex hull. Then, we can iteratively find the next point on the convex hull. Therefore, finding the optimal bit allocations is equal to repeatedly finding a vector $\underline{l}$ that satisfies (75) for every vector $\underline{v}$, where an initial vector $\underline{k}$ is given. When the desired bit rate or the desired distortion is reached, the iteration can be stopped.

The sum of (75) will be larger or equal to zero if all the terms are zero. The objective is to find a vector $\underline{l}$ such that $\forall v_n$

$$[r_n(v_n)-r_n(k_n)]-S(\underline{l},\underline{k})[d_n(v_n)-d_n(k_n)] \geq 0. \tag{76}$$

Two different situations are obtained depending on the sign of $d_n(v_n)-d_n(k_n)$, each resulting in one of the two neighbors of $\underline{k}$

$$d_n(v_n)-d_n(k_n)<0 \Rightarrow S(\underline{l},\underline{k}) \geq \frac{r_n(v_n)-r_n(k_n)}{d_n(v_n)-d_n(k_n)} \quad \forall v_n \tag{77}$$

$$d_n(v_n)-d_n(k_n)>0 \Rightarrow S(\underline{l},\underline{k}) \leq \frac{r_n(v_n)-r_n(k_n)}{d_n(v_n)-d_n(k_n)} \quad \forall v_n. \tag{78}$$

Let us consider the first case. The solution to (77) is [119]

$$S(\underline{l},\underline{k}) = \max_{n=1}^{N} s(v_n,k_n) \quad \forall v_n \tag{79}$$

where $s(v_n,k_n)$ is defined by

$$s(v_n,k_n) = \frac{r_n(v_n)-r_n(k_n)}{d_n(v_n)-d_n(k_n)}. \tag{80}$$

Now, by substituting (70) and (71) into the definition of $S(\underline{l},\underline{k})$, and applying the definition of $s(v_n,k_n)$, we obtain

$$S(\underline{l},\underline{k}) = \frac{\sum_{n=1}^{N} s(l_n,k_n)[d_n(l_n)-d_n(k_n)]}{\sum_{n=1}^{N}[d_n(l_n)-d_n(k_n)]}. \tag{81}$$

Now, by using (79) we can modify (81) to

$$\sum_{n=1}^{N}\left[s(l_n,k_n)-\max_{n=1}^{N} s(v_m,k_m)\right][d_n(l_n)-d_n(k_n)] = 0. \tag{82}$$

The key of the algorithm is in the solution of this equation. The trivial solution $\underline{l}=\underline{k}$ is not considered, and the solution is

$$\begin{aligned} l_n &= v_n \quad \text{for that } n \text{ for which } s_n(v_n,k_n) \text{ is maximum} \\ l_n &= k_n \quad \text{for all other } n. \end{aligned} \tag{83}$$

Valid solutions of (83) are those satisfying the first condition of (77), since we considered this case. This means concretely that we are searching in the direction of decreasing distortion D. This is the sought-after solution since very low bit-rate coding of images is the objective. Therefore, it is reasonable to begin the search with the lowest possible bit rate. If the objective is very low distortion, then it is of interest to begin the search with the lowest distortion, the second condition of (77) has to be developed in a similar way.

Now, all the tools of the algorithm have been developed. The algorithm can be summarized as follows.

Algorithm 1 *Optimal Bit Allocation*

1 Determine the initial bit allocation by assigning to each sub-band an admissible quantizer that has the lowest bit rate.
2 INITIALIZE ← TRUE.
3 **do**
4 **if** INITIALIZE **then**
5 Calculate for each sub-band n all possible values of $s(v_n, k_n)$ with $d(v_n) < d_n(k_n)$ and find for each sub-band the quantizer for which $s(v_n, k_n)$ is maximal.
6 INITIALIZE ← FALSE.
7 **else**
8 Calculate the values $s(v_n, k_n)$ for the subband to which the new quantizer was assigned and find the quantized for which $s(v_n, k_n)$ is maximal.
9 Determine the sub-band for which the maximum $s(v_n, k_n)$ is the largest and assign to this sub-band the quantizer for which this maximum is obtained.
10 Calculate the new rate R and the new distortion D.
11 **until** The bit rate is sufficiently close to the desired bit rate.

IX. IMAGE COMPRESSION STANDARDS

A. JPEG

JPEG [12], [124] has been created out of the collaboration between the Consultative Committee on International Telephone and Telegraph (CCITT) and the International Standards Organization (ISO) to establish an international standard for image compression. It is based on transform coding using the DCT.

The original image is divided into 8×8 blocks which are separately transformed. Note that the correlation between different blocks is not exploited. After transformation, the 64 transform coefficients are quantized by different quantization steps. These are specified by a quantization matrix to take into account the different importance of each coefficient. Quantization steps for low-frequency coefficients are smaller than those for high-frequency coefficients. The coefficients are then source coded using either Huffman coding or arithmetic coding.

The partitioning of the image into 8×8 blocks is not a source of distortion by itself; however, the independent quantizing of image blocks introduces a block effect. It is especially visible on image regions with low local variance.

B. JPEG Lossless

In some applications, lossless compression of a still image is of prime importance. For these applications, the JPEG group has also proposed a lossless variant. It is based on a completely different approach than the lossy algorithm described previously [12]. It does not make use of any transform. The redundancy between neighboring pixels is removed by means of a simple prediction. The value of a pixel is first predicted from one or several of its neighbors (seven patterns can be used to predict the value of a pixel.) The prediction error is coded by means of a Huffman code. This compression technique achieves a modest compression ratio of about 2 : 1 for natural images. It also suffers from the lack of progressive decoding.

In fact, a bit stream generated by lossless mode of JPEG cannot be decoded partially to allow user to have a glance of the picture. The major advantage of lossless JPEG is its extreme simplicity.

C. FlashPix

Some of the drawbacks of the current JPEG standard have been resolved in the new format called FlashPix [125]. FlashPix is more of a new format than a new compression scheme. In most imaging applications, the user does not need to have an immediate access to a high resolution of the whole image. Therefore, FlashPix stores several resolutions of the same image. The user can choose one of them for intermediate or final processing stages. With this multiresolution format, the appropriate resolution can be selected. On a low-resolution monitor, a low-resolution image will be used, and when printing the final result, a higher resolution version of the image is selected. In FlashPix, each resolution level is compressed using the JPEG algorithm.

D. MPEG-4

MPEG is working on a technology for the compression of audio-visual data in multimedia environment. Video is object based in MPEG-4 and normally deals with sequences of images; however, its intracoding mode is a still image compression scheme. The algorithm is very similar to that of JPEG. The bounding box of the object to be coded is divided into 16×16 macroblocks, containing four blocks of 8×8 pixels. A DCT is performed separately on each of the luminance and chrominance blocks in a given macroblock, totaling six 8×8 blocks as the chrominance is usually subsampled. Three types of macroblocks may be encountered in a bounding box: those that lie completely inside the object; those completely outside of the object; and those that partially cover the object. Macroblocks that are completely outside of the video object plane (VOP) are not coded at all. Those macroblocks that lie completely inside the object are coded using a conventional DCT scheme. The 8×8 blocks of a macroblocks lying partially on the VOP are first padded using repetitive padding prior to their coding. Padding gives a value to pixels which are outside of the boundary of the object in order to perform block matching (e.g., for motion estimation and compensation).

The DCT coefficients are quantized, zig-zag scanned, and entropy coded by run-length and Huffman methods. DCT coefficients are split into two groups: zero frequency coefficient (DC) and nonzero frequency coefficients (AC). Two quantization techniques are available. The first technique, similar to that of recommendation H.263 [109], uses a

quantization parameter for the AC coefficients in the DCT. This value may change based on a desired quality or a targeted bit rate, on a macroblock by macroblock basis. The second technique for quantization is similar to that used in MPEG-2 [108], where the quantization step size may vary depending on the position of the AC coefficient in the frequency domain according to a quantization matrix. Intra- and interquantization matrices can be defined and can also be encoded in the bit stream. In all cases, the DC coefficients are quantized with a step size equal to eight. For a higher compression efficiency, the DC as well as the first row or first column of the AC coefficients in the intracoded macroblocks can be differentially coded using an adaptive predictive method. The prediction value is selected as the corresponding value of a neighboring block. The DC gradient between blocks is used to select the prediction neighborhood. For objects of arbitrary shape, the shape information is also coded by means of a bitmap coding approach.

X. Comparative Results

A. Introduction

The complete coding schemes described in Section IX can find several applications. In this section, two main applications are discussed and results of the described coding schemes are presented. Also, the functionalities of the described coding schemes are discussed and illustrated.

The first category of applications is the lossy compression of natural images at low to medium bit rates. This type of compression is typically required for multimedia applications or for the distant access of images, such as displaying images over Internet. In this type of compression, a visual distortion of the image is accepted in order to obtain a high compression ratio. This high compression ratio allows either a small storage space for the image or a considerable reduction of the transfer time to display the image from a distant server. The critical issue in this type of compression is the characteristic of the introduced distortion.

The second category of applications is the compression of images requiring a lossless representation of the reconstructed image. A lossless compression is often required for medical applications, as a diagnostic must be drawn on the original image. This type of compression does not allow a high compression ratio; however, the browsing capability is a very efficient way of saving time when the pictures are located on a distant server. Indeed, a lossless representation of the image is certainly not necessary on all the images which are displayed, as a lossy representation might suffice for browsing purpose. This functionality can be very useful if the user likes to visualize rapidly several pictures and deciding only afterwards if he wants the full resolution of it. Therefore, the possibility of progressively decoding a picture adds a useful functionality to these compression schemes.

B. Lossy Compression

1) Introduction: The schemes described in Section IX are used to assess their performance for lossy image compression of natural images at medium bit rates. They are compared to state-of-the-art techniques from literature. They are: the EZW algorithm proposed by Shapiro [43]; the EPIC compression scheme proposed by Simoncelli *et al*. [126], [127]; the AFB scheme proposed by Egger [44]; and the ASD compression [58], the TCQ method, and a fractal-based encoding technique.

2) Image Quality Measure: The best instrument to measure image quality is the human eye. Unfortunately, visual tests are expensive to perform, and other techniques to measure an image quality are required.

The simplest way of measuring the coding distortion is termed the peak signal-to-noise ratio (PSNR). This form of measurement is also the most used in practice because of its simplicity. It is defined as follows:

$$\mathrm{PSNR} = 10 \log_{10} \frac{255^2}{\sigma_\varepsilon^2} \tag{84}$$

where σ_ε^2 is the variance of the error given by

$$\sigma_\varepsilon^2 = \frac{1}{N} \sum_{i=0}^{N-1} (x_i - \hat{x}_i)^2 \tag{85}$$

where x_i is a sample of the original image, $\hat{x}_i$ is the same sample of the reconstructed image, and N is the number of considered pixels. This measure does not model the human visual system (HVS) [128] well. For example, it does not reflect visual artifacts such as blockiness or ringing artifacts in their real values. An image suffering from blocking artifact enjoys a poor visual quality by definition since these artifacts will render the coded image irksome to the end user.

Intensive researches have been devoted to the definition of an image distortion measure correlated with the HVS [129], [128]. In order to measure the coding performance of the different coding schemes, we will measure its PSNR as well as its visual decibels, or vdB's.

The unit vdB is a visual quality measure that takes into account mechanisms of the HVS such as contrast sensitivity (strong contrast may modify the perception of shapes), masking (closely coupled stimuli in the spectral domain can render each other imperceptible), and multichannel structure (the same pattern is not perceived the same way if moving or oriented differently). The measure is based on a Gabor decomposition with both spatial and temporal filters, leading to a perceptual decomposition of the input images consisting of channels. These channels are then processed by a masking evaluator, and the visual quality metric is then a weighted sum of the distortion measures in each channel. This last number is measured in vdB [129].

3) Coding Results: To assess the compression performance of the different schemes, three different test images have been compressed at rates where the distortion becomes visible. Fig. 40 shows the picture "Lena" compressed at 0.10 bits/pixel with the different coding schemes. One can observe the different distortions introduced by the coding schemes. For the picture compressed with the international standard JPEG, a strong blocking effect is introduced. The

Fig. 40. "Lena" 512 × 512 pixels, 0.10 bits/pixel: (a) original; (b) JPEG; (c) EPIC; (d) EZW; (e) AFB; (f) ASD; (g) TCQ; and (h) fractal.

Fig. 41. "Pepper" 512 × 512 pixels, 0.10 bits/pixel: (a) original; (b) JPEG; (c) EPIC; (d) EZW; (e) AFB; (f) ASD; (g) TCQ; and (h) fractal.

Table 3
"Lena" 512 × 512 Pixels, 0.10 bits/pixel; PSNR and *vdB* of the Different Coding Schemes

Coding scheme	JPEG	EPIC	EZW	AFB	ASD	Fractal	TCQ
PSNR	24.14	27.46	27.87	27.69	26.05	26.84	30.25
vdB	43.61	50.86	50.75	50.66	45.84	49.33	55.18

two SBC schemes EPIC and EZW have a much better visual quality than the picture compressed with JPEG; however, they suffer from another artifact, the ringing effect. One can clearly see this artifact around the contours of the image. Although less annoying than the blocking effect, it disturbs the overall quality of the coded image. The coding scheme based on the linear subband decomposition using the AFB filter bank clearly has less ringing effect in comparison to the two other SBC schemes. The picture compressed with the ASD enjoys a complete absence of ringing effect, but with a less accurate texture rendition than the pictures coded with the linear subband decompositions. Fractal-encoded images preserve most sharp edges and do not present any ringing of blocking effect, but they suffer from blurriness.

Table 4
"Pepper" 512 × 512 Pixels, 0.10 bits/pixel; PSNR and vdB of the Different Coding Schemes

Coding scheme	JPEG	EPIC	EZW	AFB	ASD	Fractal	TCQ
PSNR	24.90	27.10	27.37	27.73	26.72	26.19	29.37
vdB	47.70	51.20	50.82	51.24	51.41	49.32	54.65

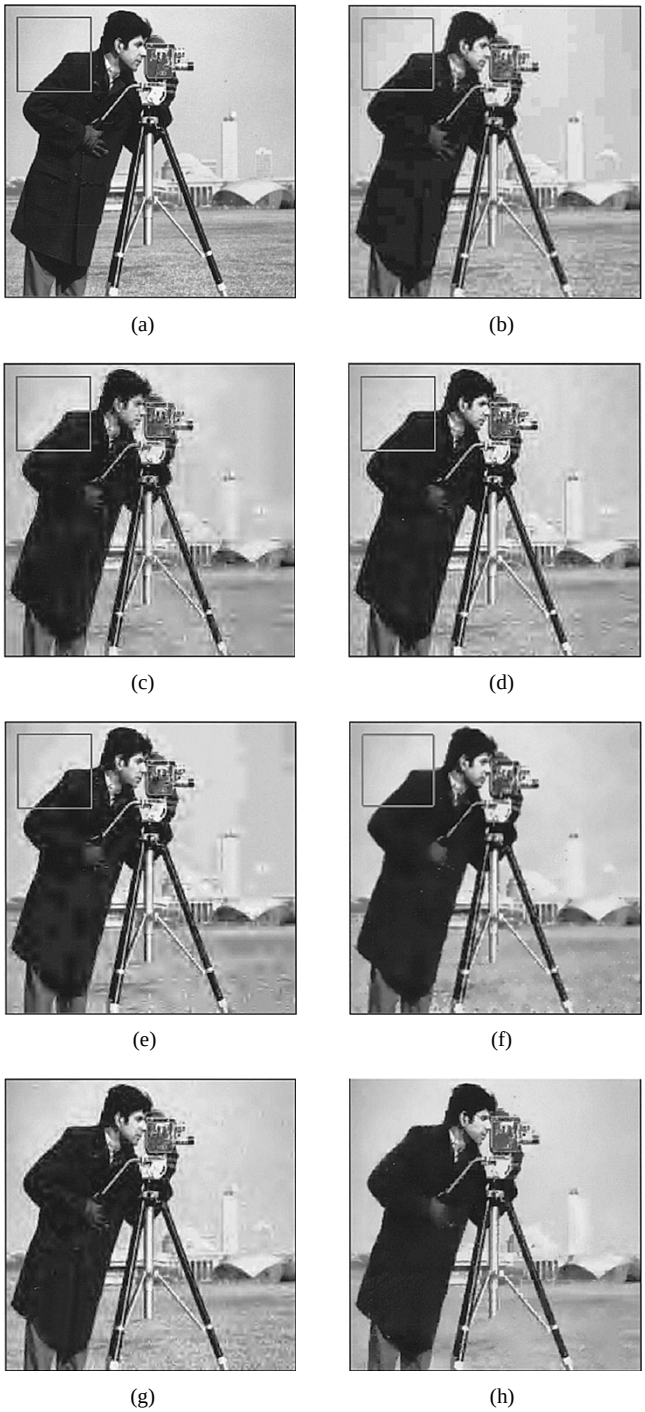


Fig. 42. "Camera Man" 256 × 256 pixels, 0.35 bits/pixel: (a) original; (b) JPEG; (c) EPIC; (d) EZW coding scheme; (e) generalized EZW with AFB filters and noise reduction; (f) ASD; (g) TCQ; and (h) fractals.

Compression with TCQ has a quite good preservation of edges, but ringing effect is present and shaded areas have stepwise color quantization. For comparison purposes, the PSNR and the vdB are shown in Table 3. The pictures

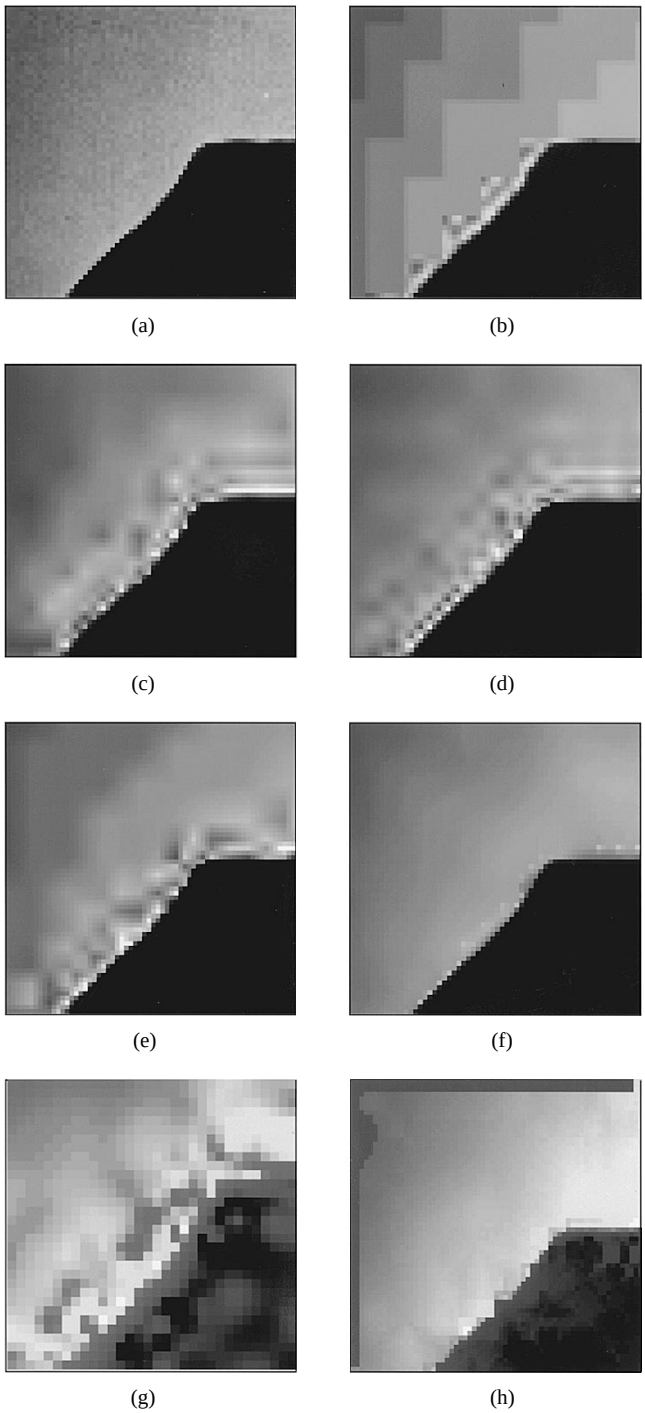
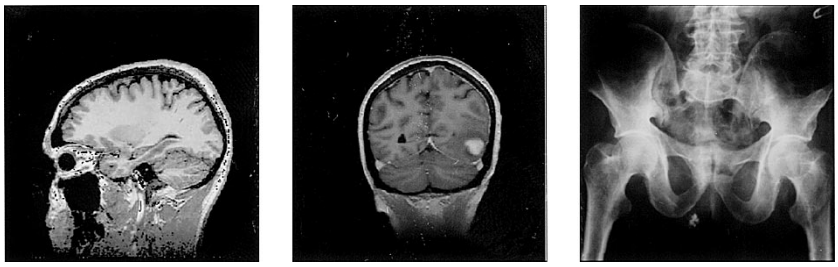


Fig. 43. Zoom of the "Camera man" 256 × 256 pixels, 0.35 bits/pixel: (a) original; (b) JPEG; (c) EPIC; (d) EZW; (e) AFB; (f) ASD; (g) TCQ; and (h) fractal.

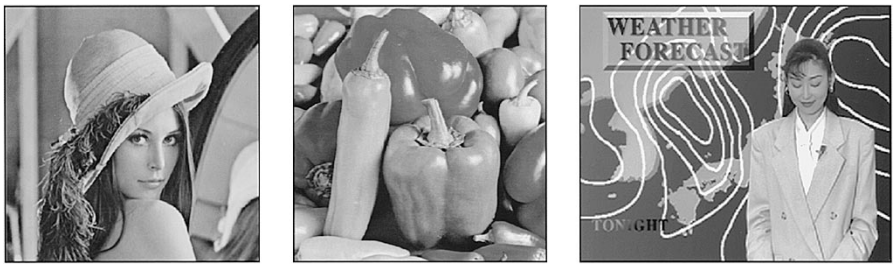
having the highest PSNR are the coding scheme EZW and the coding scheme based on the AFB's. From a visual point of view, however, the two best coding schemes are the linear SBC scheme AFB and the adaptive coding scheme based on the ASD. Among the two, it is difficult to clearly state which one is better.

Fig. 41 shows the comparison on the picture "Pepper" at 0.10 bits/pixel. This picture has very few textured regions but many sharp contours. The same artifacts as for the the



(a) (b) (c)

Fig. 44. The three medical test images: (a) "Sagittal," of type MRI; (b) "Coronal," of type MRI; and (c) "Pelvis," of type X ray.



(a) (b) (c)

Fig. 45. The three natural test images: (a) "Lena"; (b) "Pepper"; and (c) "Weather."

picture "Lena" appear in all coding schemes. The JPEG compression standard suffers from a strong blocking effect while the linear SBC schemes suffer from a ringing effect. One can clearly observe that there is a strong attenuation of the ringing effect with the AFB coding scheme in comparison with the other two classical SBC scheme. The ASD enjoys a complete absence of ringing effect. Fractals have the same blurring effect but keep edges clear, and TCQ presents ringing effect while preserving edges quite well. Table 4 compares the PSNR and the vdB of the coded images. The scheme having the highest PSNR is the AFB scheme while JPEG has the smallest PSNR. From a visual point of view, however, the ASD has the best performance and the AFB scheme comes out as second.

Fig. 42 shows the comparison on the picture "Camera Man" at 0.35 bits/pixel. This image is characterized by sharp contours around the man, by a large homogeneous region defining the sky, and by a textured region representing the ground. One can clearly observe strong artifacts on the contours of the man. There is a ringing effect with the linear SBC scheme and TCQ, and a mosquito noise with the JPEG scheme. No ringing effect occurs with the ASD scheme, nor with fractals. A zoom of a region containing a sharp contour is shown in Fig. 43, which emphasizes the good behavior of the ASD on sharp contours. Fractals preserve the contours quite well, but the gray values are not very accurate, as show in the zoomed pictures. The sky is well represented with all the coding schemes, except for JPEG, where a strong blocking effect arises, and TCQ, which quantizes the color scale in a visible way. On the other hand, in the region depicting the ground, a good representation is obtained with JPEG and the linear coding schemes, while the ASD scheme exhibits a poor reconstruction in this region, and fractals blurred out most of the details of the background.

C. Lossless Compression

In this section, two different methods for lossless compressions are compared. The first one is based on a multiresolution approach based on either the MSD decomposition [130] or the rank-order polynomial decomposition (ROPD) [131].

The second one is the lossless mode of the international standard JPEG. The lossless mode of JPEG allows seven different predictors. For each test image, the best predictor is selected for the purpose of a fair comparison.

The possibility of data browsing gives another functionality to the proposed coding schemes. This is achieved thanks to the progressive bit stream of the lossless compression. Results illustrating the browsing quality are shown and comparisons are drawn between the two proposed decompositions.

Finally, the application of lossless compression of arbitrarily shaped regions is shown for ultrasound images. In this type of image, only a part of the image is of medical interest. Hence, only this part needs to be coded in a lossless way. Results of this hybrid lossy/lossless compression are reported.

Test images of different types have been used for comparison purposes. Three medical images have been used. Two of them come from medical resonance imaging (MRI) and one is of type X ray. The three medical images are shown in Fig. 44. Three images of a general type are compared as well. They are shown in Fig. 45.

Table 5
Compression Comparison of MSD, ROPD, JPEG, SPITH, and WTCQ

Image	Size	Type	JPEG	MSD	ROPD	SPITH	WTCQ
Medical:			b/p	b/p	b/p	b/p	b/p
"Sagittal"	256 × 256	MRI	4.37	3.67	3.40	3.45	4.15
"Coronal"	256 × 256	MRI	2.13	1.45	1.39	1.52	4.89
"Pelvis"	448 × 448	X-ray	2.69	2.47	2.32	2.38	2.68
General:			b/p	b/p	b/p	b/p	b/p
"Lena"	512 × 512	Natural	4.70	4.43	4.33	4.20	4.42
"Pepper"	512 × 512	Natural	4.99	4.76	4.66	4.63	4.74
"Weather"	352 × 288	Natural	4.92	4.72	4.52	4.29	4.99

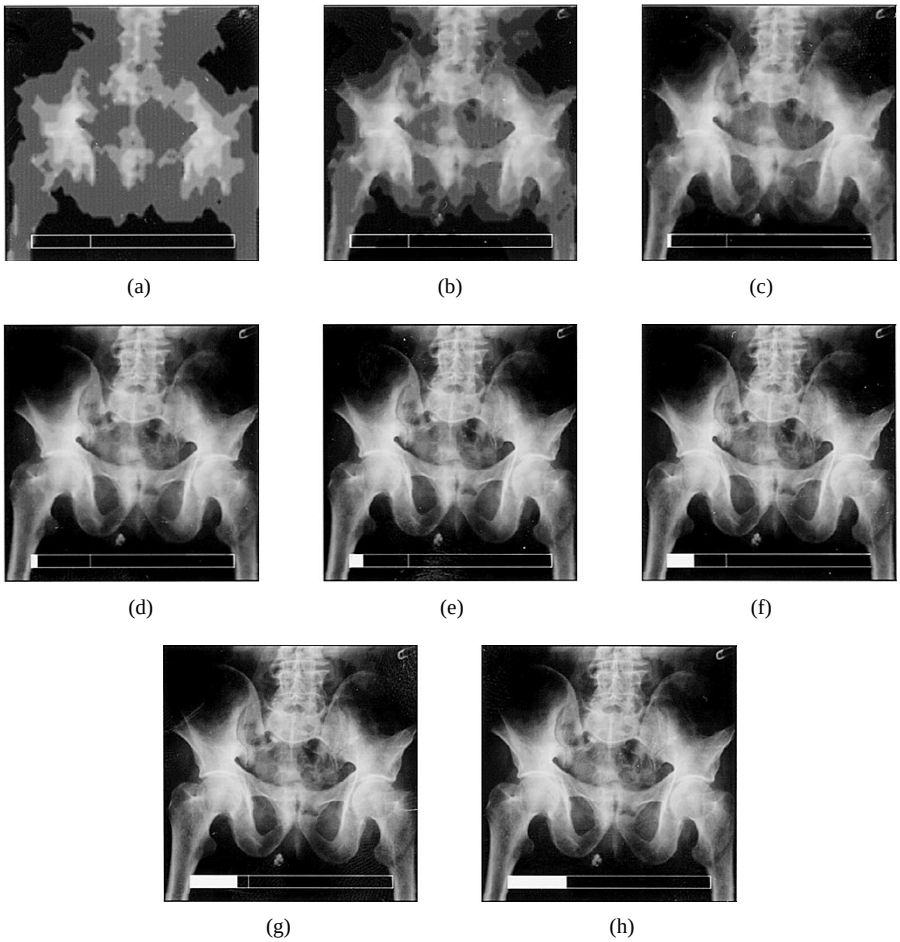


Fig. 46. "Pelvis" 448 × 448 pixels, lossless rate 2.32 bits/pixel. Each picture has been decoded with only part of the bit stream: (a) with 1.3% of the bit stream, 24.29 dB; (b) 2.6%, 28.49 dB; (c) 5.6%, 33.61 dB; (d) 11.2%, 38.58 dB; (e) 22.4%, 43.21 dB; (f) 44.8%, 46.33 dB; (g) 81.0%, 55.97 dB; and (h) original picture: 100%, ∞ dB. The full bar represents the amount of information used to reproduce the image, the line represents the lossless rate, and the empty bar represents the amount of information of the uncompressed original picture.

All test images have been compressed in a lossless way with the two decompositions (MSD and ROPD) and with the standard JPEG. The results are summarized in Table 5. It is shown that for all the test images, the multiresolution algorithms perform better than the standard JPEG. An improvement of up to 53% is achieved for MRI type of images. From the table one can also clearly see the marked superiority of the ROPD in comparison with the MSD. For natural images, an improvement of around 9% is achieved with the ROPD in comparison with the standard JPEG.

1) Illustration of the Progressive Bit Stream: The ROPD scheme has the functionality of having a completely embedded bit stream. That means that a picture compressed in a lossless way can be decompressed at any bit rate. In order to recognize the picture, only a small part of the bit stream is necessary. This is illustrated in Fig. 46 for the X-ray image "Pelvis" using the ROPD. It is demonstrated that only a small part of the bit stream is necessary to get a good quality picture. With only 5% of the bit stream, the picture is of good browsing quality. Fig. 47 shows the R-D

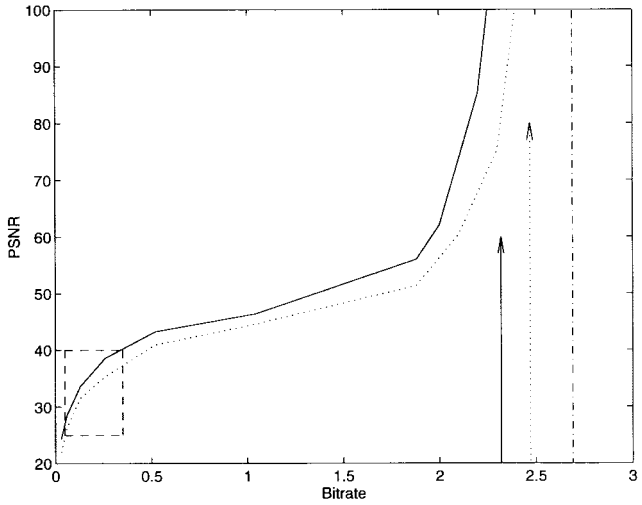


Fig. 47. R-D curve of the progressive lossless schemes for the picture "Pelvis." (Full lines) ROPD, the arrow represents the lossless rate. (Dotted lines) MSD, the arrow represents the lossless rate. (Dash dotted line) JPEG lossless rate.

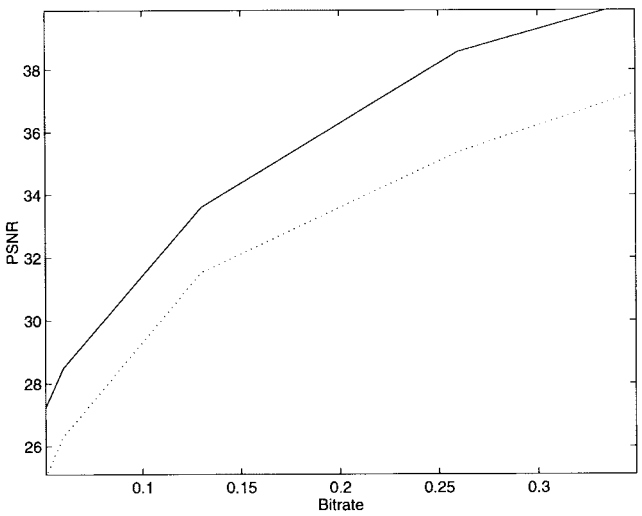


Fig. 48. Zoom of the R-D curve of the progressive lossless schemes for low bit rates for the picture "Pelvis." (Full line) ROPD. (Dotted line) MSD.

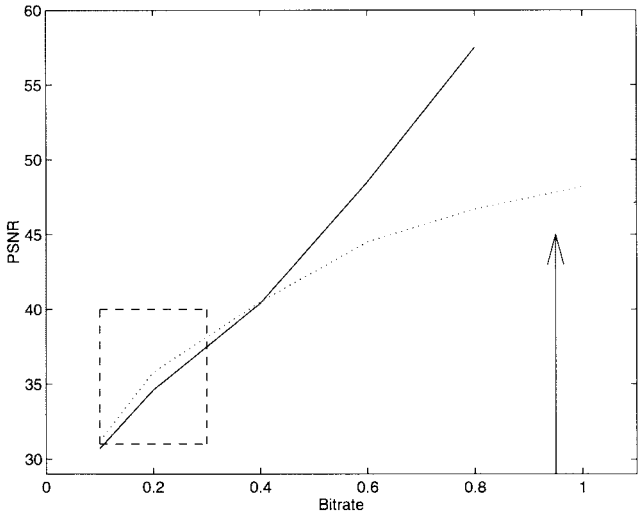


Fig. 49. R-D curve of the progressive lossless compression in comparison to a classical lossy subband compression. The arrow represents the lossless bit rate and is 1.69 bits/pixel.

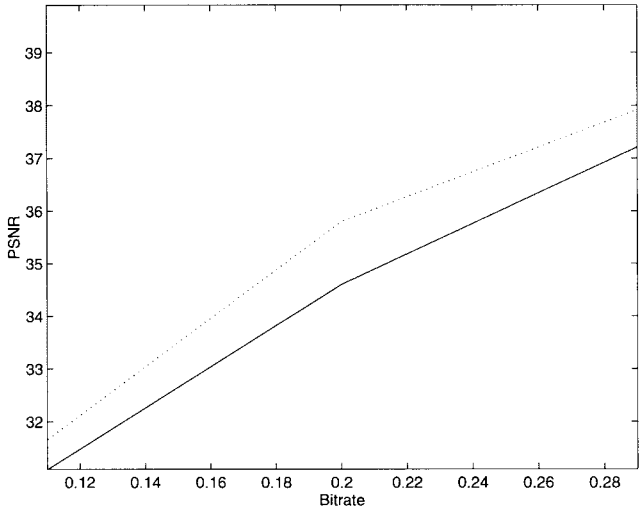


Fig. 50. R-D curve of the progressive lossless compression in comparison to a classical lossy subband compression at low bit rates.

curve of the two multiresolution coding schemes. It can be observed on the figure that the quality is increasing with an increasing bit rate. Also, as expected, the PSNR has to tend to ∞ at the lossless bit rate. The lossless rates are represented in the graph by the two arrows. One can see that the ROPD is superior to the MSD at any bit rate. At low bit rates the improvement with the ROPD is around 3 dB. This is illustrated in Fig. 48, where a zoom of the previous figure is shown. That means that the ROPD has a better lossless rate than the MSD but also has a much better browsing quality if decompressing only part of the bit stream.

In Fig. 49, the browsing quality of the progressive lossless coding is compared to the AFB coding scheme. It can be observed that the linear SBC scheme has a better performance at low bit rates than the lossless coding scheme. At high rates, however, one can see that the lossless coding scheme takes advantage of the fact that it will become lossless at a certain rate and shows a better performance than the lossy linear coding scheme. Fig. 50 shows a zoom of the R-D curve at low bit rates. One can observe that a loss of around 1 dB occurs in comparison to the linear SBC scheme. This loss represents the cost of the lossless functionality.

XI. Conclusions

The goal of Part I of this twofold paper has been to review state of the art still image coding techniques. The major building blocks of image coding schemes are overviewed.

All image-compression schemes rely on two major processing steps: message extraction and codeword assign-

ment. Message-extraction methods have been classified into predictive, fractal, block transform, and multiresolution approaches. Predictive methods are suited to lossless and low-compression applications. Fractals take advantage of self similarities in images. Transform-based coding schemes achieve higher compression ratios for lossy compression but suffer from blocking artifacts at high compression ratios. Multiresolution approaches are suited for lossy as well as for lossless compression. At lossy high-compression ratios, the typical artifact in the reconstructed images is the ringing effect.

New applications in a multimedia environment have raised the need of new functionalities of the image coding schemes. For that purpose, second-generation coding techniques segment the image into semantically meaningful parts. Therefore, parts of these methods have been adapted to work for arbitrarily shaped regions. In order to add another functionality, which is the progressive transmission of the information, specific quantization algorithms must be defined. A final step in the compression scheme is achieved by the codeword assignment.

Coding results have been presented which compare state-of-the-art techniques for lossy and lossless compression. The different artifacts for the lossy techniques are highlighted. Block-based transform coding suffers mainly from the so-called blocking artifact. It is shown that the human visual system is especially sensitive to this kind of image distortion. While they do not suffer from any blocking artifact, SBC schemes introduce another distortion called ringing effect. It is shown that this artifact can be strongly reduced by an appropriate design of the filter bank.

Results of lossless compression are also reported. It is shown that multiresolution approaches can greatly outperform predictive methods such as used in the standard JPEG. Furthermore, they allow the definition of additional functionalities, such as progressive transmission. It is shown that only 5% of the lossless bit stream is necessary to reconstruct a good browsing quality of an image.

ACKNOWLEDGMENT

The authors wish to extend special thanks to J. Reichel and C. Le Buhan-Jordan for all their helpful hints and revisions, and also to M. W. Marcellin and A. Bilgin from the Signal Processing and Coding Laboratory (SPACL) of the University of Arizona for producing the TCQ coded images.

REFERENCES

[1] M. Kunt, A. Ikonomopoulos, and M. Kocher, "Second-generation image coding techniques," *Proc. IEEE*, vol. 73, pp. 549–574, Apr. 1985.
[2] M. Kunt, M. Bénard, and R. Leonardi, "Recent results in high-compression image coding," *IEEE Trans. Circuits Syst.*, vol. 34, pp. 1306–1336, Nov. 1987.
[3] A. Ikonomopoulos and M. Kunt, "High compression image coding via directional filtering," *Signal Processing*, vol. 8, no. 2, pp. 179–203, Apr. 1985.
[4] M. Kocher and M. Kunt, "Image data compression by contour texture modeling," in *Proc. SPIE Int. Conf. Applications of Digital Image Processing*, Geneva, Switzerland, Apr. 1983, pp. 131–139.
[5] S. Carlsson, "Sketch-based coding of grey level images," *Signal Processing*, vol. 15, no. 1, pp. 57–83, July 1988.
[6] X. Ran and N. Farvardin, "Low bit-rate image coding using a three-component image model," Univ. Maryland, College Park, MD, Tech. Rep. TR 92-75, 1992.
[7] ——, "Adaptive DCT image coding on a three-component image model," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, San Francisco, CA, vol. III, Mar. 1992, pp. 201–204.
[8] S. G. Mallat and S. Zhong, "Characterization of signals from multiscale edges," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, pp. 710–732, July 1992.
[9] J. Froment and S. G. Mallat, *Second Generation Compact Image Coding With Wavelets*. New York: Academic, 1992.
[10] N. D. Memon and K. Sayood, "Lossless image compression: A comparative study," in *Proc. SPIE*, vol. 2418, 1995, pp. 8–20.
[11] S. Wong, L. Zaremba, D. Gooden, and H. K. Huang, "Radiologic Image Compression—A review," *Proc. IEEE*, vol. 83, pp. 194–219, Feb. 1995.
[12] JTC1 Commitee, "Digital compression and coding of continuous-tone still images," Int. Org. Standardization ISO/IEC, JTC1 Commitee Draft, JPEG 8-R8, 1990.
[13] V. K. Heer and H. K. Reinfelder, "A comparison of reversible methods for data compression," in *Proc. SPIE*, vol. 1233, 1990, pp. 354–365.
[14] G. R. Kuduvali and R. M. Rangayyan, "Performance analysis of reversible image compression techniques for high-resolution digital teleradiology," *IEEE Trans. Med. Imag.*, vol. 12, pp. 430–445, Sept. 1993.
[15] N. Memon, S. Ray, and K. Sayood, "Differential lossless encoding of images using nonlinear predictive techniques," in *Proc. Int. Conf. Image Processing (ICIP)*, Washington, DC, vol. III, 1994, pp. 841–844.
[16] N. Tavakoli, "Lossless compression of medical images," in *Proc. 4th Annu. IEEE Symp. Computer-Based Medical Systems*, 1991, pp. 201–207.
[17] M. F. Barnsley, *Fractals Everywhere*. San Diego, CA: Academic, 1988.
[18] A. E. Jacquin, "Image coding based on a fractal theory of iterated contractive image transformations," *IEEE Trans. Image Processing*, vol. 1, pp. 18–30, Jan. 1992.
[19] Y. Fisher, "A discussion of fractal image compression," in *Chaos and Fractals*, Saupe D. H. O. Peitgen, H. Jurgens, Eds. New York: Springer-Verlag, 1992, pp. 903–919.
[20] G. E. Oien, "L2-optimal attractor image coding with fast decoder convergence," Ph.D. dissertation, Norwegian Univ. Sci. Technol., Trondheim, Norway, 1993.
[21] E. W. Jacobs, Y. Fisher, and R. D. Boss, "Image compression: A study of iterated transform method," *Signal Processing*, vol. 29, pp. 251–263, Dec. 1992.
[22] K. Barthel, T. Voyé, and P. Noll, "Improved fractal image coding," in *Proc. Picture Coding Symp. (PCS)*, Lausanne, Switzerland, Mar. 1993, no. 1.5.
[23] A. K. Jain, "Image transforms," in *Fundamentals of Digital Image Processing* (Prentice Hall Information and System Sciences Series). Englewood Cliffs, NJ: Prentice-Hall, 1989, ch. 5.
[24] H. Karhunen, "Über lineare methoden in der wahrscheinlichkeits-rechnung," *Ann. Acad. Science Fenn*, vol. A.I. 37, 1947.
[25] M. Loève, "FIR and IIR analysis/synthesis systems for subband image coding, chapter 3," in *Processus Stochastiques et Mouvement Brownien*, P. Evy, Ed. Paris, France: Hermann, 1948.
[26] A. K. Jain, "A Fast Karhunen Loève transform for a class of random processess," *IEEE Trans. Commun.*, vol. COM-24, pp. 1023–1029, Sept. 1976.
[27] M. J. Narasimha and A. M. Peterson, "On the computation of the discrete cosine transform," *IEEE Trans. Commun.*, vol. COM-26, pp. 934–936, June 1978.
[28] W. B. Pennebaker and J. L. Mitchell, *JPEG Still Image Data Compression Standard*. New York: Van Nostrand Reinhold, 1993.
[29] H. Caglar, Y. Liu, and N. Akansu, "Optimal PR-QMF design for subband image coding," *J. Visual Commun. Image Representation*, vol. 4, no. 4, pp. 242–253, Sept. 1993.
[30] R. E. Crochiere, S. A. Webber, and F. L. Flanagan, "Digital

coding of speech in sub-bands," *Bell Syst. Tech. J.*, vol. 55, no. 8, pp. 1069–1085, 1976.
[31] M. J. T. Smith and T. P. Barnwell, "Exact reconstruction techniques for tree structured subband coders," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-34, pp. 434–441, June 1986.
[32] P. P. Vaidyanathan, "Quadrature mirror filter bank, M-band extensions and perfect reconstruction technique," *IEEE Acoustics, Speech, Signal Processing Mag.*, vol. 4, pp. 1035–1037, July 1987.
[33] ——, "Theory and design of M channel maximally decimated QMF with arbitrary M, having perfect reconstruction property," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-35, pp. 476–496, Apr. 1987.
[34] M. Vetterli and D. Le Gall, "Perfect reconstruction FIR filter banks: Some properties and factorization," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 37, pp. 1057–1071, July 1989.
[35] M. Vetterli, "Multi-dimensional subband coding: Some theory and algorithms," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-32, pp. 97–112, Apr. 1984.
[36] J. Woods and S. O'Neil, "Subband coding of images," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-34, pp. 1278–1288, Oct. 1986.
[37] J. D. Johnston, "A filter family designed for use in quadrature mirror filter banks," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, Apr. 1980, pp. 291–294.
[38] K. Nayebi, T. P. Barnwell, and M. J. T. Smith, "Time domain conditions for exact reconstruction in analysis/synthesis systems based on maximally decimated filter banks," in *Proc. Southeastern Symp. System Theory*, Mar. 1987, pp. 498–503.
[39] ——, "Time domain filter bank analysis: A new design theory," *IEEE Trans. Signal Processing*, vol. 40, pp. 1412–1429, June 1992.
[40] P. P. Vaidyanathan, T. Q. Nguyen, Z. Doganata, and T. Saramaki, "Improved technique for design of perfect reconstruction FIR QMF banks with lossless polyphase matrices," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-27, pp. 1042–1056, July 1979.
[41] D. Field, "Relation between the statistics of natural images and the response properties of cortical cells," *J. Opt. Soc. Amer.*, vol. 4, no. 12, pp. 2379–2394, Dec. 1987.
[42] A. N. Akansu, R. A. Haddad, and H. Caglar, "The binomial QMF-wavelet transform for multiresolution signal decomposition," *IEEE Trans. Signal Processing*, vol. 41, pp. 13–19, Jan. 1993.
[43] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, pp. 3445–3462, Dec. 1993.
[44] O. Egger and W. Li, "Subband coding of images using asymmetrical filter banks," *IEEE Trans. Image Processing*, vol. 4, pp. 478–485, Apr. 1995.
[45] W. Li and O. Egger, "Improved subband coding of images using unequal length PR filters," in *Proc. 14th Gretsi Symp. Signal and Image Processing*, Juan-les-Pins, France, Sept. 1993, pp. 451–454.
[46] R. M. Haralick, X. Zhuang, and J. Lee, "The digital morphological sampling theorem," *IEEE Trans. Acoustic, Speech, Signal Processing*, vol. 37, pp. 2067–2089, Dec. 1989.
[47] M. Kunt, *Traitement numérique des signaux, Traité d'Electricité* (in French). Lausanne, Switzerland: Presses Polytechniques Romandes, 1984; *Digital Signal Processing* (in English). Norwood, MA: Artech House, 1986.
[48] P. Salembier and M. Kunt, "Size-sensitive multiresolution decomposition of images with rank order based filters," *Signal Processing*, vol. 27, no. 2, pp. 205–241, May 1992.
[49] Z. Zhou and A. N. Venetsanopoulos, "Morphological methods in image coding," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 3, Mar. 1992, pp. 481–484.
[50] J. R. Casas, L. Torres, and M. Jare no, "Efficient coding of residual images," in *Proc. SPIE, Visual Communications Image Processing*, vol. 2094, 1993, pp. 694–705.
[51] A. Toet, "A morphological pyramid image decomposition," *Pattern Reconstruction Lett.*, vol. 9, no. 4, pp. 255–261, May 1989.
[52] M. J. T. Smith and T. P. Barnwell, "Exact reconstruction techniques for tree structured subband coders," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-34, pp. 434–441, June 1986.
[53] H. Blume and A. Fand, "Reversible and irreversible image data compression using S-transform and Lempel–Ziv coding," in *Proc. SPIE*, vol. 1091, 1989, pp. 2–18.
[54] G. Deng, "A new interpolative subband coding algorithm for lossless image compression," in *Proc. Int. Conf. Image Processing (ICIP)*, Lausanne, Switzerland, vol. I, Sept. 1996, pp. 93–96.
[55] H. Jung, T. Choi, and R. Prost, "Rounding transform for lossless coding," in *Proc. Int. Conf. Image Processing (ICIP)*, Lausanne, Switzerland, vol. I, Sept. 1996, pp. 65–68.
[56] P. Roos, M. A. Viergever, M. A. A van Dikje, and J. H. Peters, "Reversible intraframe compression of medical images," *IEEE Trans. Med. Imag.*, vol. 7, pp. 328–336, Dec. 1988.
[57] L. Wang and M. Goldberg, "Comparative performance of pyramid data structures for progressive of medical imagery," in *Proc. Medical Imaging IV: Image Capture and Display, SPIE*, vol. 1232, 1990, pp. 403–413.
[58] O. Egger, W. Li, and M. Kunt, "High compression image coding using an adaptive morphological subband decomposition," *Proc. IEEE*, vol. 83, pp. 272–287, Feb. 1995.
[59] R. Nagarajan, P. Au, M. Etoh, and T. Ankei, "Core experiment S4i on grayscale shape coding effects," Int. Standardization Org., Chicago, IL, Tech. Rep. ISO/IEC JTC1/SC29/WG11, Oct. 1996.
[60] "MPEG-4 video verification model version 5.1," Int. Standardization Org., Maceio, AL, Tech. Rep. ISO/IEC JTC1/SC29/WG11, Dec. 1996.
[61] M. Gilge, T. Engelhardt, and R. Mehlan, "Coding of arbitrarily shaped image segments based on a generalized orthogonal transform," *Signal Processing: Image Commun.*, vol. 1, no. 2, pp. 153–180, Oct. 1989.
[62] H. H. Chen, M. R. Civanlar, and B. G. Haskell, "A block transform coder for arbitarily shaped image segments," in *Proc. Int. Conf. Image Processing (ICASSP)*, Austin, TX, vol. I, pp. 85–89, Nov. 1994.
[63] T. Sikora, "Low complexity shape-adptive DCT for coding of arbitrarily shaped image systems," *Signal Processing: Image Commun.*, vol. 7, pp. 381–395, Nov. 1995.
[64] T. Sikora and B. Makai, "Shape-adaptive DCT for generic coding of video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, pp. 59–62, Feb. 1995.
[65] D. C. Youla, "Mathematical theory of image restoration by the method of convex projections," in *Image Recovery: Theory and Applications*. New York: Academic, 1987.
[66] C. L. Buhan Jordan, S. Bhattacharjee, F. Bossen, F. Jordan, and T. Ebrahimi, "Shape representation and coding of visual objects in multimedia applications—An overview," *Ann. Telecommun.*, to be published.
[67] J. Foley, A. Van Dam, S. Feiner, and J. Hughes, Eds., *Computer Graphics: Principles and Practice*. Reading, MA: Addison-Wesley, 1987.
[68] D. H. Ballard and C. M. Brown, *Computer Vision*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1982.
[69] "Facsimile coding schemes and coding control functions for group 4 facsimile apparatus," ITU-T, Geneva, Switzerland, ITU-T Recommendation T.6.
[70] N. Yamaguchi, T. Ida, and T. Watanabe, "A binary shape coding method using modified MMR," in *Proc. Int. Conf. Image Processing (ICIP)*, Oct. 1997, vol. 1, pp. 504–507.
[71] F. Bossen and T. Ebrahimi, "A simple and efficient shape coding technique based on bitmap representation," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, 1997, vol. 4, pp. 3129–3132.
[72] N. Brady, F. Bossen, and N. Murphy, "Context-based arithmetic encoding of 2D shape sequences," in *Proc. Int. Conf. Image Processing (ICIP)*, 1997, pp. 29–32.
[73] L. Shapiro, R. MacDonald, and S. Sternberg, "Ordered structural shape matching with primitive extraction by mathematical morphology," *Pattern Recognition*, vol. 20, no. 1, pp. 75–90, Feb. 1987.
[74] P. Brigger, "Morphological shape representation using the skeleton decomposition: Application to image coding," Ph.D. dissertation, EPFL, Lausanne, Switzerland, 1995.
[75] H. Freeman, "On the encoding of arbitrary geometric configuration," *IRE Trans. Electron. Comput.*, vol. EC-10, pp. 260–268, June 1961.
[76] C. R. Dyer, A. Rosenfeld, and H. Samet, "Region representa-

tion: Boundary codes from quadtrees," *Commun. ACM*, vol. 23, no. 3, pp. 171–179, Mar. 1980.
[77] H. Samet, "Region representation: Quadtrees from boundary codes," *Commun. ACM*, vol. 23, no. 3, pp. 163–170, Mar. 1980.
[78] M. Eden and M. Kocher, "On the performance of a contour coding algorithm in the context of image coding Part I: Contour segment coding," *Signal Processing*, vol. 8, no. 10, pp. 381–386, 1985.
[79] C. Lu and J. G. Dunham, "Highly efficient coding schemes for contour lines based on chain code representation," *IEEE Trans. Commun.*, vol. 39, pp. 1511–1514, Oct. 1991.
[80] T. Kaneko and M. Okudaira, "Encoding of arbitrary curves based on chain code representation," *IEEE Trans. Commun.*, vol. 33, pp. 697–707, July 1985.
[81] R. D. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.
[82] U. Ramer, "An iterative procedure for polygonal approximation of plane curves," *Comput. Graphics Image Processing*, vol. 1, pp. 244–256, Feb. 1972.
[83] J. Dunham, "Optimum uniform piecewise linear approximation of planar curves," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 8, pp. 67–75, Jan. 1986.
[84] K. W. Stuhlmuller, A. Salai, and B. Girod, "Rate-constrained contour representation for region-based motion compensation," in *Proc. Symp. Visual Communications and Image Processing, SPIE*, vol. 2727, Mar. 1996, pp. 344–355.
[85] C. Gu, "Multivalued morphology and segmentation-based coding," Ph.D. dissertation, EPFL, Lausanne, Switzerland, 1996.
[86] P. Gerken, "Object-based analysis-synthesis coding of image sequences at very low bit rates," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, pp. 228–235, June 1994.
[87] F. Moscheni, F. Dufaux, and H. Nicolas, "Entropy criterion for optimal bit allocation between motion and prediction error information," in *Proc. SPIE's Conf. Visual Communications and Image Processing (VCIP)*, 1993, pp. 235–242.
[88] E. Reusens, "Joint optimization of representation model and frame segmentation for generic video compression," *Signal Processing Image Commun.*, vol. 46, pp. 105–117, Sept. 1995.
[89] T. Breger, "Optimal quantizers and permutation codes," *IEEE Trans. Inform. Theory*, vol. 18, pp. 759–765, Nov. 1972.
[90] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, July 1948.
[91] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84–95, Jan. 1980.
[92] R. M. Gray, "Vector quantization," *IEEE Acoustics, Speech, Signal Processing Mag.*, vol. 1, pp. 4–29, Apr. 1984.
[93] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell, MA: Kluwer, 1992.
[94] I. Moccagatta and M. Kunt, "Lattice vector quantization approach to image coding," in *SPIE Proc. Visual Communications and Image Processing '92*, Boston, MA, Nov. 1992, pp. 430–440.
[95] I. Moccagatta, "Image coding and notion field segmentation using vector quantization," Ph.D. dissertation, EPFL, Lausanne, Switzerland, 1995.
[96] G. Ungerboek, "Channel coding with multilevel/phase signals," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 55–67, Jan. 1982.
[97] M. W. Marcellin and T. R. Fischer, "Trellis coded quantization of memoryless and Gauss-Markov sources," *IEEE Trans. Commun.*, vol. COM-38, pp. 82–93, Jan. 1990.
[98] A. J. Viterbi and J. K. Omura, "Trellis encoding of memoryless discrete-time sources with a fidelity criterion," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 325–332, May 1974.
[99] H. S. Wang and N. Moayeri, "Trellis coded vector quantization," *IEEE Trans. Commun.*, vol. 40, pp. 1273–1276, Aug. 1992.
[100] J. W. Owens, M. W. Marcellin, B. R. Hunt, and M. Kleine, "Compression of synthetic aperture radar phase history data using trellis coded quantization techniques," in *Proc. ICIP'97*, Oct. 1997, pp. 592–595.
[101] T. R. Fischer, M. W. Marcellin, and M. Wang, "Trellis-coded vector quantization," *IEEE Trans. Inform. Theory*, vol. 37, pp. 1551–1566, Nov. 1991.
[102] J. Semenlilli, A. Bilgin, F. Shen, M. W. Marcellin, and J. C. Kieffer, "Progressive transmission in trellis coded quantization-based image coders," in *Proc. ICIP'97*, Oct. 1997, pp. 588–591.
[103] K. L. Bowles, *Problem Solving Using PASCAL*. New York: Springer-Verlag, 1977.
[104] T. M. Cover and J. A. Thomas, *Elements of Information Theory* (Wiley Seres in Telecommunications). New York: Wiley, 1991.
[105] R. N. Williams, *Adaptive Data Compression*. Norwell, MA: Kluwer, 1993.
[106] J. G. Cleary and I. H. Witten, "Data compression using adaptive coding and partial string matching," *IEEE Trans. Inform. Theory*, vol. 32, pp. 396–402, July 1984.
[107] "Information technology—Coding of moving pictures and associated audio for digital storage media up to about 1.5 Mbit/s—Part 2: Coding of moving picture information," Int. Org. Standardization ISO/IEC, Tech. Rep. JTC1 CD 11172,1991.
[108] "Information theory—Generic coding of moving pictures and associated audio information—Part 2: Video," Int. Org. Standardization ISO/IEC, Tech. Rep. DIS 13818-2, 1994.
[109] "Recommendation H.263—Video coding for narrow telecommunication channels at <64 kbit/s," ITU-T, Geneva, Switzerland, Tech. Rep., July 1995.
[110] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proc. IRE*, vol. 40, no. 9, pp. 1098–1101, 1952.
[111] "MPEG-4 video verifiction model versiou 7.0," Int. Org. Standardizatin ISO/IEC, Tech. Rep. JTC1/SC29/WG11, Dec. 1996.
[112] L. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Commun. ACM*, vol. 30, pp. 520–540, June 1987.
[113] A. Moffat, R. Neal, and I. H. Witten, "Arithmetic coding revisited," in *Proc. Data Compression Conf.*, 1995, pp. 202–211.
[114] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*. Englewood Cliffs, NJ: Prentice-Hall, 1984.
[115] A. N. Akansu and Y. Liu, "On-signal decomposition techniques," *Opt. Eng.*, vol. 30, pp. 912–920, July 1991.
[116] H. Caglar and A. N. Akansu, "A generalized parametric PR-QMF design technique based on Bernstein polynomial approximation," *IEEE Trans. Signal Processing*, vol. 41, pp. 2314–2321, July 1993.
[117] D. Le Gall and A. Tabatabai, "Subband coding of digital images using symmetric short kernel filters and arithmetic coding techniques," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, 1988, pp. 761–764.
[118] J. Katto and Y. Yasuda, "Performance evaluation of subband coding and optimization of its filter coefficients," in *Proc. SPIE, Visual Communications and Image Processing*, vol. 1605, 1991, pp. 95–106.
[119] P. H. Westerink, J. Biemond, and D. E. Boekee, "An optimal bit allocation algorithm for subband coding," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, 1988, pp. 757–760.
[120] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Optimal pruning with applications to tree-structured source coding and modeling," *IEEE Trans. Inform. Theory*, vol. 35, pp. 299–315, Mar. 1989.
[121] E. A. Riskin, "Optimal bit allocation via the generalized BFOS Algorithm," *IEEE Trans. Inform. Theory*, vol. 37, pp. 400–402, Mar. 1991.
[122] H. Everett, "Generalized lagrange multiplier method for solving problems of optimum allocation of resources," *Oper. Res.*, vol. 11, pp. 399–417, 1963.
[123] G. T. Toussaint, "Pattern recognition and geometrical complexity," in *Proc. 5th Int. Conf. Pattern Recognition*, Miami, FL, Dec. 1980, pp. 1324–1347.
[124] G. K. Wallace, "The JPEG still picture compression standard," *Commun. ACM*, vol. 34, no. 4, pp. 31–34, Apr. 1991.
[125] C. R. Hauf and J. S. Houchin, "The FlashPix image file format," in *Proc. IS&T/SID 4th Color Imaging Conf.: Color Science, Systems and Applications*, 1996, Springfield, VA, pp. 234–238.
[126] E. P. Simoncelli and E. H. Adelson, *Subband Tranforms,*. Norwell, MA: Kluwer, 1990, ch. 4.
[127] E. H. Adelson and E. P. Simoncelli, "Subband image coding with tree-tap pyramids," in *Proc. Picture Coding Symp. (PCS)*, Cambridge, MA, 1990.
[128] C. J. van den Branden Lambrecht, "Perceptual models and architectures for video coding applications," Ph.D. dissertation, EPFL, Lausanne, Switzerland, Sept. 1996.
[129] S. Comes, "Les traitements perceptifs d'images numerisées," Ph.D. dissertation, Univesité Catholique de Louvain, France, 1995.

[130] O. Egger and M. Kunt, "Embedded zerotre based lossless image coding," in *Proc. Int. Conf. Image Processing (ICIP)*, Washington, DC, vol. III, pp. 616–619, 1995.
[131] O. Egger, "Region representation using nonlinear techniques with applications to image and video coding," Ph.D. dissertation, EPFL, Lausanne, Switzerland, 1997.

Olivier Egger was born in Fribourg, Switzerland, on November 12, 1969. He received the diploma in electrical engineering in 1994 and the Ph.D. degree in 1996 from the Swiss Federal Institute of Technology (EPFL), Lausanne.

He spent one year at the Georgia Institute of Technology as an exchange student, where he specialized in signal processing. He then joined the Signal Processing Laboratory of EPFL, where he specialized in filter bank theory and design, morphological signal processing, image segmentation, and image coding. He currently heads his own company, Oasya S.A., specializing in automatic strategy engines for computer games.

Dr. Egger was awarded the Landry and SVIA of Switzerland prizes for his research in image processing.

Pascal Fleury was born in Delémont, Switzerland, on February 25, 1969. He received the M.Sc. degree in communication systems from the Swiss Federal Institute of Technology (EPFL), Lausanne, in July 1994. He is currently pursuing the Ph.D. degree at the Signal Processing Laboratory (LTS), EPFL.

He spent one year at the Eurécom Institute, Sophia Antipolis, France, where his major was multimedia. He also spent a year in the Speech Group at the IBM T. J. Watson Research Labs, Yorktown Heights, NY, where he worked in task adaptation for speech recognizers. His current interests include coding scheme selection in image compression and related aspects of algorithm complexity.

Touradj Ebrahimi was born on July 30, 1965. He received the M.Sc. and Ph.D. degrees, both in electrical engineering, from the Swiss Federal Institute of Technology (EPFL), Lausanne, in 1989 and 1992, respectively.

From 1989 to 1992, he was a Research Assistant at the Signal Processing Laboratory of EPFL. During the summer 1990, he was a Visiting Researcher at the Signal and Image Processing Institute of the University of Southern California, Los Angeles. In 1993, he was a Research Engineer at the Corporate Research Laboratories of Sony Corporation, Tokyo, where he conducted research on advanced video compression techniques for storage applications. In 1994, he served as a research consultant at AT&T Bell Laboratories working on very low bit-rate video coding. He is currently at the Signal Processing Laboratory of EPFL, where he is involved with various aspects of digital video and multimedia applications and is in charge of the Digital TV Group. His research interests are multidimensional signal processing, image processing, information theory, and coding. He is the author or the co-author of more than 50 research publications and six patents.

In 1989, Dr. Ebrahimi was the recipient of the IEEE and Swiss National ASE Award. He is a member of EURASIP.

Murat Kunt (Fellow, IEEE) was born in Ankara, Turkey, on January 16, 1945. He received the M.S. degree in physics and the Ph.D. degree in electrical engineering from the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, in 1969 and 1974, respectively.

From 1974 to 1976, he was a Visiting Scientist at the Research Laboratory of Electronics of the Massachusetts Institute of Technology, where he developed compression techniques for X-ray images and electronic image files. In 1976, he returned to EPFL, where he is presently Professor of Electrical Engineering and Director of the Signal Processing Laboratory. He teaches and conducts research in digital signal and image processing with applications to modeling, coding, pattern recognition, scene analysis, industrial developments, and biomedical engineering. He also consults for governmental offices, including the French General Assembly. He is the author or co-author of more than 200 research papers and 14 books, and he hold three patents.

Dr. Kunt is a founding member of EURASIP, the European Association for Signal Processing. He has served as chairman and/or a member of the scientific committees of several international conferences and on the editorial boards of the PROCEEDINGS OF THE IEEE, *Pattern Recognition Letters*, and *Traitement du Signal*, and he was the Editor-in-Chief of the *Signal Processing Journal*. He was the co-chairman of the first European Signal Processing Conference, which was held in Lausanne in 1980, and the General Chairman of the International Image Processing Conference (ICIP'96) held in Lausanne in 1996. He was the President of the Swiss Association for Pattern Recognition from its creation until 1997. He received the gold medal of EURASIP for meritorious services and the IEEE ASSP Technical Achievement Award in 1983 and 1997, respectively.