Title
Wavelet zerotree image compression with packetization

Permalink
https://escholarship.org/uc/item/5rx4p5wp

Journal
IEEE Signal Processing Letters, 5(5)

ISSN
1070-9908 1558-2361

Authors
Rogers, J. K.
Cosman, P. C.

Publication Date
1998-05-01

DOI
10.1109/97.668942

Peer reviewed

Vector Quantization of Image Subbands: A Survey

Pamela C. Cosman, *Member, IEEE*, Robert M. Gray, *Fellow, IEEE*, and Martin Vetterli, *Fellow, IEEE*

Invited Paper

***Abstract*—Subband and wavelet decompositions are powerful tools in image coding because of their decorrelating effects on image pixels, the concentration of energy in a few coefficients, their multirate/multiresolution framework, and their frequency splitting, which allows for efficient coding matched to the statistics of each frequency band and to the characteristics of the human visual system. Vector quantization (VQ) provides a means of converting the decomposed signal into bits in a manner that takes advantage of remaining inter and intraband correlation as well as of the more flexible partitions of higher dimensional vector spaces. Since 1988, a growing body of research has examined the use of VQ for subband/wavelet transform coefficients. We present a survey of these methods.**

I. Introduction

IMAGE compression maps an original image into a bit stream suitable for communication over or storage in a digital medium. The number of bits required to represent the coded image should be smaller than that required for the original image so that one can use less storage space or communication time. There are two basic types of compression. *Lossless* compression, which is also called noiseless coding, data compaction, entropy coding, or invertible coding, refers to algorithms that allow the original pixel intensities to be perfectly recovered from the compressed representation. *Lossy* compression algorithms do not allow that. This paper is concerned with *digital* compression techniques, and if an image is analog in space and amplitude to begin with, one must first render it discrete in both space and amplitude. Discretization in space is generally called sampling. This consists of examining the intensity of the analog image on a regular grid of points called *picture elements* or *pixels*. Discretization in amplitude is often called quantization. This consists of mapping a number from a continuous range of possible values into a finite set of approximating values. The term analog-to-digital (A/D) conversion is often used to mean both sampling and quantization, that is, the conversion of a signal that is analog in both space and amplitude to a signal that is discrete in both space and amplitude. Such a conversion is by itself an example of lossy compression.

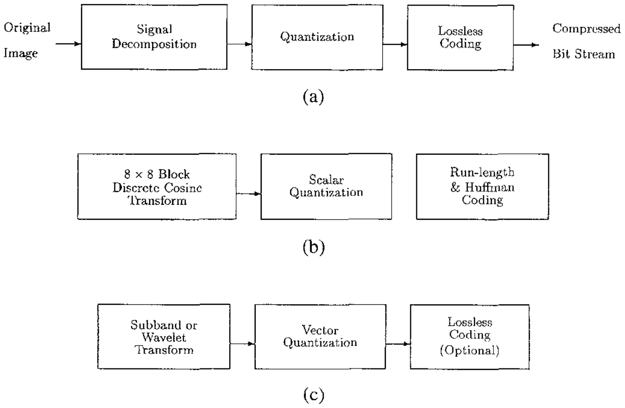


Fig. 1. Image compression systems.

A general system for digital image compression is shown in Fig. 1(a). It consists of one or more of the following operations, which may be combined with each other or with additional signal processing:

- *Signal decomposition*: The image is decomposed into several images for separate processing, typically by linear transformation by a Fourier or discrete cosine transform or by filtering with a subband or wavelet filter bank. The goal is to concentrate energy in a few coefficients, to reduce correlation, or to provide a useful data structure.
- *Quantization*: High-rate digital pixel intensities are mapped into a relatively small number of symbols. This operation is nonlinear and noninvertible; it is "lossy." The conversion can operate on individual pixels (scalar quantization (SQ)) or groups of pixels (vector quantization (VQ)). Quantization can include throwing away some of the components of the signal decomposition step.
- *Lossless compression*: Further compression is achieved by an invertible (lossless, entropy) code such as Huffman, Lempel-Ziv, or arithmetic code. The idea here is to assign codewords with a few bits to likely symbols and codewords with more bits to unlikely symbols so that the average number of bits is minimized.

A bewildering variety of image compression systems have been proposed, which involve various choices for each of the three basic components. The JPEG still-image compression

Manuscript received July 21, 1995; revised August 1, 1995. This work was supported by a postdoctoral fellowship from the National Science Foundation, by ARPA grant F49620-93-1-0558 administered by AFSOR, by the Army Medical Research and Materiel Command under Grant DAMD17-94-J-4354, by the National Science Foundation under Grant no. MIP-931190, by NSF Award MIP-93-21302, and by the California State Micro program in collaboration with HP Laboratories, Philips, Sun Microsystems, and Tektronix.

P. C. Cosman is with the Department of Electrical and Computer Engineering, University of California at San Diego, San Diego, CA 92037 USA.

R. M. Gray is with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305 USA.

M. Vetterli is with the Dpartment of Electrical Engineering and Computer Science, University of California at Berkeley, Berkeley, CA 94720 USA.

Publisher Item Identifier S 1057-7149(96)01410-8.

standard, for example, uses a discrete cosine transform for the first step, SQ with different quantizer step sizes for the different transform coefficients in the second step, and run-length coding combined with Huffman coding for the third step, as shown in Fig. 1(b). This paper is concerned with those image compression systems that use a subband or wavelet decomposition for the first step, some form of VQ for the second step, and any lossless coding (or none at all) for the third step, as shown in Fig. 1(c). We also briefly consider systems based on scalar quantization (SQ) because of its simplicity and historical importance. It also provides comparisons with the systems using VQ. Although SQ can be viewed as a special case of VQ with 1-D vectors, many systems using SQ are uniquely suited to 1-D vectors and do not immediately generalize to larger dimensions.

This paper treats only subband/VQ systems used for image compression. VQ can be used for many other types of image processing besides compression [1], and subband/VQ systems can similarly be used for noncompressive purposes as well. For example, wavelet/VQ systems have been used for signature verification [2], for automatic target recognition from high-range resolution radar returns [3], and other clustering and classification uses. This survey paper will not include these other uses of subband/VQ.

The organization of the paper is as follows. In Section II, we present a brief introduction to subband and wavelet image coding systems. We begin with a description of the subband decomposition. The reader is expected to be already somewhat familiar with this material, and we include it primarily to establish the notation we will use for the remainder of the paper. For thorough introductions to subband and wavelet coding, the reader is referred to a plethora of books, tutorials, and survey articles [4]–[9]. Section II also discusses some issues of entropy coding, SQ, and bit allocation as they apply to subband image coding systems. Section III presents a brief overview of several VQ techniques that have been studied in the context of subband coding. Section IV presents a survey of a wide variety of subband/VQ systems. Last, Section V discusses differences between VQ applied to subband coefficients and VQ applied directly to image pixels, and offers some conclusions.

II. Subband Systems for Image Coding

A. Subband Decompositions

Subband coding was introduced in the context of speech coding in 1976 by Croisier *et al.* [10] and Crochiere *et al.* [11]. Croisier *et al.* were the first to solve the critical problem of aliasing cancellation after decimation and reconstruction in subbands, using "quadrature mirror filters" (QMF's). The theoretical extension of subband filtering from 1-D to 2-D was made by Vetterli [12]. Two-dimenisonal subband filtering was first applied to image coding by Woods and O'Neil [13]. A subband decomposition is produced by an analysis filter bank followed by downsampling. Any or all of the resulting subbands can be further input to an analysis filter bank and downsampling operation for as many stages as desired. Fig. 2 shows one stage of such a system using 2-D separable filters. In the figure, ↓ 2 denotes downsampling by a factor of two, H_0 denotes a lowpass filter, and H_1 denotes a highpass filter. The initial high and lowpass filters and downsampling are applied to the rows of an image. The subsequent filters and downsampling are then applied to the resulting columns. Because there are only two filters, this is called a *two-channel* system. Multichannel systems have been much less explored than two-channel systems. The image is split into four bands—LL, LH, HL, and HH—according to whether the rows and columns received the low- or high-frequency filtering. The reconstruction operation consists of an upsampling operator followed by a synthesis filter bank, as shown in Fig. 3. One stage of reconstruction is used for each stage of decomposition. In this paper, we use the term "decomposition" to refer to the filtering and downsampling operations for as many stages as desired.

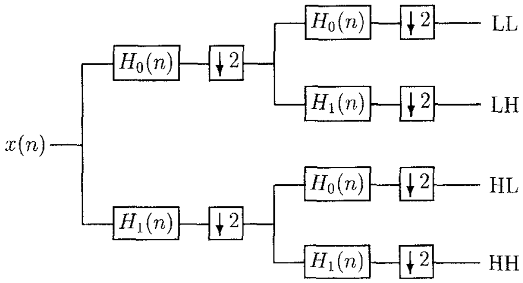


Fig. 2. One stage of subband decomposition with 2-D separable filters.

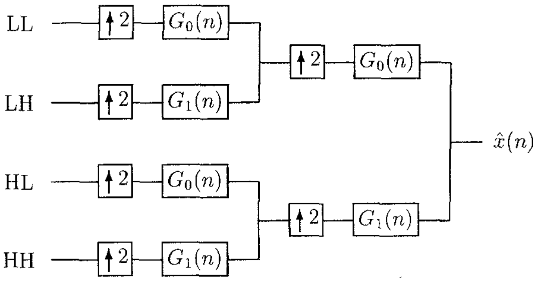


Fig. 3. One stage of subband reconstruction with 2-D separable filters.

Ideally, decomposition followed by reconstruction will provide a perfect replica of the original signal. This perfect reconstruction requirement is usually built into the decomposition as a requirement if real inputs and analog arithmetic are assumed, but it is only an approximation when dealing with digital signals because of round-off error in the arithmetic. Furthermore, we are here interested in image compression, in which case the decomposed signal is compressed in a lossy fashion, typically by SQ or VQ. The term "encoding" refers to the combination of decomposing and quantizing with or without subsequent entropy coding. The term "decoding" refers to the inverse operation to encoding; therefore, it consists of the combination of inverse quantization followed by reconstruction.

If *all* four of the subbands are subjected to another stage of filtering and downsampling, this leads to a *uniform* decomposi-

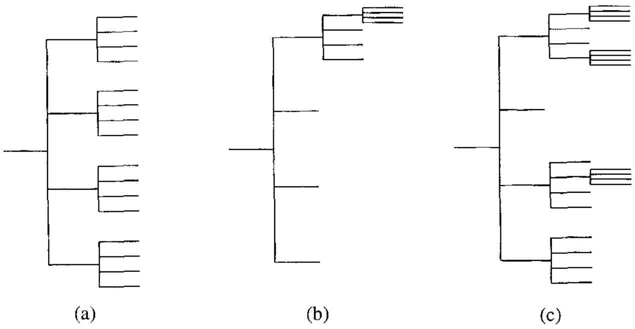


Fig. 4. (a) Two stages of uniform decomposition leading to 16 uniform subbands; (b) three stages of pyramidal decomposition leading to 10 octave-band subbands; (c) wavelet packet.

tion, as depicted in Fig. 4(a). This decomposition is a balanced tree. If *only* the low band is further decomposed, this is referred to as an *octave-band* decomposition (which is also often called a *logarithmic-tree* or pyramid[1] decomposition). This situation, which is represented by a highly unbalanced tree, is shown in Fig. 4(b). In fact, for the fixed maximum depth of the tree, this octave-band decomposition is as unbalanced a tree as one can have. For the fixed maximum depth of the tree, the balanced tree produced by a uniform decomposition and the maximally unbalanced tree produced by a pyramidal decomposition represent the two extremes of possible choices. Anything between these extremes is called a *wavelet packet* [15]. An example of a wavelet packet is shown in Fig. 4(c).

Techniques for producing image subband decompositions can be quite general. One type of decomposition, called a wavelet transform, uses octave-band filter banks with regularity conditions imposed on the lowpass filters. The wavelet transform can be interpreted as a subband decomposition of the signal into a set of overlapping frequency channels having the same bandwidth on a logarithmic scale. In this paper, we will refer to this type of octave-band decomposition as a wavelet decomposition. The more general term "subband decomposition" will be used for the uniform-band case and as the general term.

The various subband/VQ systems surveyed used a wide variety of filters. One of the goals of signal decomposition theory is to find conditions such that in the absence of noise or quantization, the overall reconstruction is perfect or nearly so. The simplest and shortest filter—the Haar wavelet—is important for historical and theoretical reasons but is rarely used as part of a wavelet/SQ or wavelet/VQ scheme. By far, the most common choice is one of the quadrature mirror filters QMF's that are listed in [16]. Many authors refer to these as "Johnston's near-perfect reconstruction filters" or near-PR filters. The second most popular choice is one of the family of Daubechies orthogonal filters [17], [8]. The biorthogonal filters [18], [19], which possess the advantage of linear phase, are also common. The Gabor functions, which are Gaussians modulated by complex exponentials, are more rarely used. Use of the latter for wavelet transforms is motivated by the fact that they have joint localization in the spatial/spatial-frequency domains that is optimal in a certain sense and that, according to recent experiments, the majority of receptive field profiles of the human visual system fit quite well this type of function. The choice of filter reflects a tradeoff among many factors, such as spatial localization, frequency selectivity, regularity, and coding gain. A number of papers have examined the question of selecting filter coefficients [20]–[24], but as this survey is concerned with quantization, no more will be said on filters.

B. Entropy Coding

Let $\{Y_n\}$ be a source that produces symbols from a discrete alphabet A. In our application, Y_n is a sequence of indexes produced by quantizing (scalar or vector) another sequence X_n. Define the k-tuple probabilities $p(y^k) = \Pr(Y^k = y^k)$ for all $y^k = (y_0, y_1, \dots, y_{k-1}) \in A^k$, which is the set of all k-dimensional vectors with coordinates in A. A noiseless code for Y_n is an invertible mapping of the sequence Y_n into binary symbols V_n. The goal of noiseless coding is to have the smallest possible average number of binary symbols per original Y_n symbol. The basic strategy underlying noiseless coding is to assign codewords with a few bits (short length) to symbols or groups of symbols with high probability and codewords with more bits (long length) to unlikely symbols or groups of symbols so that the average number of bits is minimized. Approaching the optimal performance in terms of minimizing bits may conflict with the other possible design goals of the noiseless coder, which involve encoding speed, decoding speed, memory requirements, delay, and progressivity.

Noiseless codes can map individual Y_n into a varying number of bits (e.g., Huffman coding) or fixed numbers of Y_n into a varying number of bits (e.g., Huffman codes applied to vectors), or they can map variable numbers of Y_n into fixed numbers of bits (e.g., Tunstall or Lempel-Ziv codes) or varying numbers of symbols into varying numbers of bits. Shannon theory demonstrates that the smallest achievable noiseless coding bit rate for a stationary and ergodic source Y is given by the entropy rate $\overline{H}(Y)$ defined by $\overline{H}(Y) = \lim_{k\to\infty} k^{-1}H(Y^k)$, where

$$H(Y^k) = -\sum_{y^k \in A^k} p(y^k)\log_2 p(y^k).$$

This optimal performance is provably achievable in the limit of coding arbitrarily large input vectors or permitting arbitrarily large delay in variable input length codes. In the special case where $\{Y_n\}$ is a memoryless process, then $\overline{H}(Y) = H(Y_0)$, which is the so-called marginal entropy. An important point here is that simple Huffman coding on individual symbols is not guaranteed to perform close to the optimum. The optimum can be achieved to arbitrary precision only if one codes groups of increasing size.

If the original signal is decomposed and quantized, then the application of noiseless coding can provide further compression without any additional loss of information. However, if such entropy coding is to be permitted then superior perfor-

[1] Strictly speaking, a pyramid decomposition is an overcomplete representation often used in computer vision and first used for image compression by Burt and Adelson [14].

mance is obtainable in principle by designing the quantizer to provide the smallest possible entropy rate while maintaining the desired fidelity. Many of the coding systems surveyed here use entropy coding on the bit stream, and many do not. In most cases, those systems that do use entropy coding do not try to match the quantizer design with the subsequent entropy coding but merely tack the entropy coder onto the end. Those systems that include entropy coding in their reported results are pointed out as such. Whenever entropy coding is not mentioned in relation to a particular algorithm, that means the algorithm did not include entropy coding. In general, those systems that do not use entropy coding could improve their results by incorporating these techniques, although the amount of improvement is unknown and may perhaps be small. In a small number of published results, the authors report the entropy rate of the bit stream and intend this to be indicative of the achievable bit rate. As discussed above, because the bit rates of actual implementable entropy codes may or may not approach the entropy rate of the source, results reported only as entropy rates should not be placed in direct comparison with results that report actual bit rates.

C. Scalar Quantization

Quantization is the mapping of a large set of possible inputs into a smaller set of possible outputs. In SQ, the inputs are individual numbers. SQ includes such operations as "rounding to the nearest integer." The possible outputs—in this case the integers—are called *quantization levels*, *reproduction levels*, or *reconstruction levels*. Integers are evenly spaced; therefore, this is an example of *uniform* quantization. In general, the quantization levels do not have to be integers, nor are they evenly spaced. The spacing between the reproduction levels is often called the *bin width*. To specify a scalar quantizer, one needs the set of possible reproduction levels and a rule for mapping input scalars to reproduction levels.

In the simplest application of SQ to subband coefficients, one simply applies a uniform quantizer with bin width Δ to all coefficients, i.e., the real line is carved up into disjoint intervals of length Δ [25], [26]. A coefficient lying in a particular interval is represented by the midpoint (centroid with respect to Lebesgue measure) of the interval. The intervals are numbered. The coefficient is mapped into an index labeling the interval in which it lies, and these indexes are then coded using an entropy code. As Δ gets bigger, the interval size grows, the distortion increases, and the entropy decreases so that the compression is greater. As Δ gets smaller, distortion goes down, but compression is less. The resulting quantization indexes can then be entropy coded. A typical approach is to use a Peano or zigzag scan through the coefficients to form a 1-D data stream and then apply either a simple scalar Huffman or an arithmetic code to the result. Both Huffman and arithmetic codes can be combined with run-length coding to take advantage of the many zero coefficients remaining after quantization.

An alternative is to exploit both the spatial and frequency localization of wavelets. Lewis and Knowles [27] proposed to model edges across scales by gathering nonzero coefficients corresponding to a given spatial location in bands oriented in the same direction. A further development taking dependencies across scales into account is Shapiro's landmark SQ method based on embedded zerotrees [28]. This successful variation on uniform SQ attempts to take advantage of the many zeros that appear in the quantized coefficients (increasing as Δ does) and of the location correspondence between those zeros. An extension to this SQ algorithm attempted to jointly optimize the scalar quantizers and the zerotree structures by examining distortion/rate tradeoffs [29]. These methods are described further in Section IV-B together with the VQ algorithms based on them.

One way to improve on simple uniform SQ of all subbands is to adjust the quantizers to the different bands. One could use an analog of the JPEG quantization table. Nonuniform SQ's can also be used. For example, a Lloyd–Max quantizer could be designed for each subband based either on a training sequence of typical data or on a model of the data [30], [31]. With a Lloyd–Max quantizer, the quantization law is tailored to the pdf of each subband. In the discrete case, the value of the probability mass function associated with a coefficient value x is estimated by the frequency of occurrence of x in the training sequence for that subband.

Each subband can itself have a variety of SQ's available for coding different coefficients within that subband. For example, the encoder can classify the activity level of blocks of coefficients within the band and transmit that class information to the decoder so that it knows which SQ to use for coding those coefficients [13]. On the other hand, the classification can be made on the baseband, which is assumed to be transmitted first, and that way, the decoder can perform the same classification and use the correct quantizer for each later coefficient without the encoder having to explicitly send the class information as overhead [32].

Another popular design is to employ quantizers with a center "dead zone," which zeros out large numbers of coefficients and makes subsequent entropy coding very efficient (see, e.g., [33]). Such a method requires a "significance map," which indicates where the coefficients of large magnitude are located. A variety of methods, such as run-length coding, could be used for efficiently coding this sparse positional information [34].

Some works on wavelet compression place little emphasis on the quantization operation and focus almost entirely on the signal decomposition. Donoho *et al.* [35] essentially use the dead zone idea without explicitly quantizing the nonzero coefficients. Their "hard thresholding" method sets to zero all coefficients having energy less than a predetermined threshold, forming a "dead zone." The remaining coefficients are not explicitly quantized. If operating on real numbers, this method mathematically amounts to no compression at all since there is no compression in going from any number of real numbers to a much smaller collection of real numbers. On a computer, real numbers are, of course, approximated by floating point numbers. Thus, there will be some compression, but it is not immediately quantifiable since the location of the nonthresholded coefficients has to be sent as well. Thus, the scheme presented in [35] is not yet a compression scheme. A simple approach would be to explicitly quantize the nonthresholded coefficients, for example, with a uniform quantizer having a stepsize of the order of the threshold. The error would, of course, increase proportionally to the number of remaining coefficients, which we assume is small. However, the locations of these coefficients must be sent as side information with an overhead of the order of $\log_2 N$ bits for a signal of size N unless some prior information is known.

Donoho *et al.* also consider a "soft thresholding" approach that corresponds to attenuating all of the coefficient values according to some weighting function instead of making the "hard" comparison and zeroing. Again, quantization is not considered in their scheme, but the same discussion as above can be used to arrive at a realistic compression scheme.

Note that the classes of signals considered by Donoho *et al.* contain many discontinuities (e.g. like images), and thus, the wavelet basis is indeed an interesting choice because of its ability to focus on edges. For other classes of signals (e.g. containing sinusoidal components like musical sounds), that advantage is not clear.

Some subband/SQ methods attempt to compute the quantization steps of the different subbands to minimize a noise power weighted by the sensitivity function of the human visual system [36], [37]. This is a very important line of investigation since perceptually based quantization will outperform quantization based on mean squared error (MSE) by a large margin, especially at low bit rates.

D. Bit Allocation

Bit allocation is the problem of assigning bit rates to a number of sources that are to be encoded. In the case of subband coding systems, the different sources may be the different subbands but not necessarily. The goal of the allocation is typically to minimize the overall distortion of the coder subject to constraints such as a maximum overall bit rate, where the overall distortion can be defined in different ways. If the quantization is to be scalar without subsequent entropy coding, then there is the constraint that the assignment must be integer. There are a variety of other possible constraints that will be discussed later.

It is well known from experience that the low-frequency subbands have higher variance than the high-frequency subbands, and this higher variance means that they have "more information" and ought to be allocated more bits. How many more bits? A small number of researchers decided on arbitrary numbers. For example, a low-frequency band might get 2 bits per pixel (b/pixel), and a higher frequency band only 1 b/pixel, with perhaps some highest frequency bands being discarded altogether (0 b/pixel). Other researchers allocated bits to be proportional to the logarithm of the variance of each subband. In such a case, the variances would typically need to be transmitted to the decoder in order to recompute the allocation there.

The most common bit allocation method is one we will refer to as the "equal-slope" method. It is applicable to both scalar and vector quantizers. The idea is based on what is known in economics as "Pareto optimality." We assume that a variety of quantizers have been designed for each subband. For each subband, we can, in turn, apply each of the quantizers that were designed for it and compute the resulting distortion and bit rate. These distortion/rate points can be plotted in the (D, R) plane, and the lower convex hull of the set can be extracted for each band. We call these the distortion/rate curves. The "equal-slope" method, which was invented and reinvented in different contexts and from different viewpoints [38]–[43] indicates that the overall system should operate by finding a point of equal slope along the $D(R)$ curve for each band. For the given $D(R)$ curves related to those fixed VQ's, this bit allocation strategy was shown to be optimal for an orthogonal decomposition in the sense of minimizing the mean-squared error (MSE) of the reconstructed image. The intuitive explanation for this "equal-slope" method is as follows: Suppose you are operating an encoder with a variety of quantizers available to encode two sources. The variety of quantizers, which operate at different rates, yield a spread of possible distortions and rates for each source. If the channel allows you to increase your overall rate slightly, which source should you choose to increase? One source may be encoded at a flat part of its $D(R)$ curve; therefore, an increase in the rate will cause only a trivial decrease in the distortion. However, another source may be encoded on a steep part of its $D(R)$ curve so that a rate increase causes the distortion to drop sharply. One would choose to add bits to that source until it too reaches an equally flat part of its $D(R)$ curve, at which point adding an additional bit to either source would produce the same drop in distortion. A variety of further refinements to this "equal-slope" bit allocation scheme have been tried, mostly involving attempts to weight the distortion so as to reflect human visual sensitivity.

The equal-slope intuition for determining bit allocation indeed reflects asymptotic results from quantization theory, as does the older method of allocating bits proportional to log variances. It is worthwhile to describe these results briefly to provide an appreciation for their relevance and limitations. For details, see, e.g., [44, ch. 8–10] and [45]–[47].

If a single continuous random variable Y is quantized to a very high resolution or small distortion, then asymptotic approximations yield two conclusions. Denote the quantizer output by $Q(Y)$ and the average distortion by $D = E[(Y - Q(Y))^2]$.

- The optimum quantizer in the sense of minimizing the average distortion subject to a fixed (asymptotically large) number of quantization levels $N \leq 2^R$ is given approximately by $D \approx \sigma_Y^2 2^{-2R}$, where σ_Y^2 is the variance of the random variable. Equivalently, for a given average distortion D, the optimum rate R is given by $R \approx \frac{1}{2}\log_2(\sigma_Y^2/D)$. This result is most important when entropy coding is not applied to the quantizer output.
- The optimum quantizer in the sense of minimizing the average distortion subject to a fixed output entropy $H(Y) \leq R$ is approximately achieved by a uniform quantizer with (asymptotically small) bin width Δ, yielding an average distortion of $D = \Delta^2/12$ and a marginal entropy of $H(Q(Y)) \approx h(Y) - \frac{1}{2}\log_2 \Delta^2$, where $h(Y)$ is the differential entropy of the continuous random variable Y. Thus, the rate as measured by entropy for an optimized

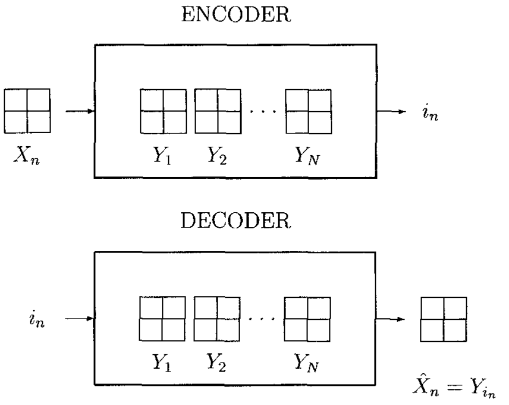


Fig. 5. Vector quantizer.

quantizer is $H(Q(Y)) \approx h(Y) - \frac{1}{2}\log_2(12D)$. This result is the appropriate approximation when subsequent entropy coding is permitted so that one wishes to trade off entropy with average distortion. Keep in mind, however, that one must code increasingly large vectors and not just the scalar Y in order to guarantee closeness to even the marginal entropy.

We can summarize these results as follows: Given asymptotically small distortion (or high rate), the optimal rate of a quantizer Q operating on a random variable Y yielding average distortion $D = E[(Y - Q(Y))^2]$ is given by $\gamma(Y) - \frac{1}{2}\log_2 D + c$, where $\gamma(Y)$ is a statistic of the random variable (half of the log of the variance if entropy coding is not used and the differential entropy if it is), and c is a constant. If the random variable is Gaussian, then the differential entropy is given by $h(Y) = \frac{1}{2}\log_2(2\pi e\sigma_Y^2)$ and, hence, is proportional to half of the log of the variance; in both cases, the optimal rate takes the form $\frac{1}{2}\log_2(\sigma_Y^2/D) + c$, where c is a constant not depending on the source or distortion. In this case, the optimal rate translates directly into an optimal bin width Δ.

These single random variable results then yield bit allocation results by simultaneously considering several random variables, say $Y_1, Y_2, \ldots, Y_K$, possibly corresponding to different transform coefficients. The bit allocation game now becomes a question of minimizing the total average distortion over all K random variables given a constraint on a total bit budget for all random variables. If R_k is the rate for the kth random variable (either $\log_2 N_k$ for an N_k level quantizer or $H(Q_k)$ for a constrained entropy quantizer) and $D_k = E[(Y_k - Q(Y_k))^2]$ is the average distortion of the kth quantizer, then the goal is to minimize the total average distortion $\sum_k D_k$ subject to an overall rate constraint $\sum_k R_k \leq R$. This problem can be easily solved by converting this constrained minimization to an unconstrained minimization using a Lagrange multiplier. The unconstrained problem is to minimize $\sum_k D_k(R_k) + \lambda \sum_k R_k$ for some $\lambda \geq 0$. This can be shown directly or by using variational principles. In the latter case, we take derivatives with respect to R_k and equate to zero: $D_k'(R_k) + \lambda = 0$ for each k. This is precisely the "equal-slope" method. Each R_k is chosen to satisfy $D_k'(R_k) = -\lambda$. In the case where $D_k(R_k) = ce^{-2R_k}$, we have $-2ce^{-2R} = -\lambda$, or $R_k = (1/2)\log(2c/\lambda)$. This is the "rule of thumb" whereby the bit allocation is an affine function of the log of the variance c.

If all of the random variables are independent and if they are such that the standardized random variables $(Y_k - E(Y_k))/\sigma_{Y_k}$ are identically distributed, then the optimal bit allocation allocates bits to Q_k proportional to $\log \sigma_{Y_k}^2$ plus some constant in the case where no entropy coding is used. The same conclusion follows for the case with entropy coding if the random variables are also assumed to be Gaussian. If the random variables do not share the same standardized distribution, then additional terms must be included.

Last, if one has correlated Y_k, then these results do not hold. If, however, the variables are jointly Gaussian and one first applies a Karhunen–Loeve transform, then one produces uncorrelated variables that are also independent and to which one can apply these asymptotic approximations and draw similar conclusions to the effect that the optimal rate allocations for both approaches are proportional to the log variance.

The key points of the preceding discussion for our case of decomposing signals by a transform (such as the wavelet transform) can now be summarized: To have a rigorous basis for the common rule of thumb that bit allocation should be made proportional to the logarithms of the coefficient variances, the following assumptions are required:

- The rate is asymptotically large (or the distortion is asymptotically small).
- The original data being transformed are jointly Gaussian.
- The transform is the Karhunen–Loeve transform.

These assumptions are often not valid in practice. The most interesting bit rates for many applications are not asymptotically large, image intensities are not well modeled by Gaussian, and wavelet transforms can at best only approximate the Karhunen–Loeve transform. In particular, wavelet transforms can render data approximately uncorrelated but not independent. Nonetheless, this approach remains common.

III. Vector Quantization

In VQ, the inputs are vectors rather than scalars. A typical VQ system is depicted in Fig. 5, where an *input vector* X_n consisting of blocks of pixels is quantized by being encoded into a binary index i_n, which then serves as an index for the output reproduction vector or codeword. If the code has a fixed rate of b b/input vector, then i_n has length b. With a variable rate code, the indexes i_n have variable length, and b is their average length. The compressed image is represented by these indexes i_n, and the compressed representation requires fewer bits than the original.

The operation of the decoder is thus completely described once we have specified the codebook. The operation of the encoder requires a choice of the mapping rule. The basic Shannon source code model provides an encoder that is optimal for a given codebook if the goal is to minimize an average distortion. If $d(X, \hat{X}) \geq 0$ measures the distortion or the cost of reproducing an input vector X as a reproduction $\hat{X}$ and if we further assume that the overall distortion (or lack of fidelity) of the system is measured by an average distortion, the optimal encoder for a given codebook selects the vector Y_i if

$d(X, Y_i) \leq d(X, Y_j)$, all j. The optimal encoder thus operates in a nearest neighbor or minimum distortion fashion. With *full-search* VQ, the encoder determines the closest codeword in its collection by an exhaustive search. A constrained search can speed up the encoding but may not be guaranteed to find the overall nearest neighbor in the codebook. The choice of distortion measure permits us to quantify the performance of a VQ in a manner that can be computed and used in analysis and design optimization. By far, the most commonly used distortion measure for image compression is the MSE, in spite of its often cited shortcomings.

There are many approaches to code design. The two most common approaches are to use some subset of a lattice to force a highly structured codebook or to apply a clustering algorithm such as the Lloyd (Forgey, Isodata, k-means) algorithm. The generalized Lloyd algorithm (GLA), which is sometimes referred to as the Linde-Buzo-Gray (LBG) algorithm, has been described in detail in a variety of places (see, for example [48], [49], [44], [50]). It iteratively improves a codebook by alternately optimizing the encoder for the decoder (using a minimum distortion or nearest neighbor mapping) and the decoder for the encoder (replacing the old codebook by generalized "centroids"). For squared error, centroids are the Euclidean mean of the input vectors mapping into a given index. Code design is almost always based on a training set of typical data rather than on mathematical models of the data. Where the GLA is mentioned in this paper, a training set should be assumed unless stated otherwise.

The GLA is a descent algorithm when run on either a probabilistic model or on a training set, and hence, it can always be used to improve a codebook in terms of reducing distortion. Many other clustering algorithms have also been applied to VQ design, including pairwise nearest neighbor (PNN) methods, Kohonen's self-organizing feature maps, simulated annealing, deterministic annealing, stochastic gradient methods, competitive learning, and other neural network methods. (Several are mentioned in [44].) Most of the relevant literature for subband coding focuses on either constrained forms of the Lloyd algorithm or on lattice codes.

Entropy-Constrained VQ (ECVQ): Ordinary VQ strives to minimize average distortion subject to a constraint on the number of codewords. This is equivalent to minimizing average distortion for a given rate, when rate is measured by the log of the number of codewords. This is a suitable optimization formulation when the system is indeed fixed rate and no subsequent entropy coding takes place. As we have argued for scalar entropy coding of quantizer outputs, if we intend to entropy code the outputs of a VQ, then it makes more sense to design the quantizer with the entropy coding in mind, that is, to minimize the average distortion subject to a constraint on the quantizer output entropy rather than on the number of codewords. An algorithm for accomplishing this was developed by Chou *et al.* [51] by modifying the GLA to include an entropy coding stage. Mathematically, this is accomplished by modifying the distortion measure from a simple squared error to include a log probability term multiplied by a Lagrangian multiplier. This results in an expected modified distortion equaling the ordinary MSE plus the quantizer output entropy. This causes the optimal encoder to replace a simple minimum squared error rule by the minimization of the squared error plus a log probability term that reflects the number of bits that will be needed to send the particular codeword.

Entropy constrained VQ can improve rate-distortion performance when a VQ is followed by an entropy coder, but the price for the improvement is added computational complexity. Vector generalizations of the asymptotic results previously described for scalar entropy coding suggest that when the rate is high (or distortion small), the optimum entropy constrained VQ is approximately the multidimensional generalization of a uniform quantizer, that is, a lattice VQ [47], [52]. Thus, the performance gains achievable by entropy constrained VQ over optimized lattice VQ are likely to be small except for very low bit rates, which are the bit rates of most interest here.

A. Vector Quantizers with Constrained Structures

Shannon theory states that VQ can perform arbitrarily close to the theoretical optimal performance for a given rate if the vectors have sufficiently large dimension. Unfortunately, unconstrained VQ is limited to fairly small vector dimensions and codebook sizes because of the search complexity. One approach to reduce this complexity is to impose constraints on the codebook. This means that the codewords cannot have arbitrary locations as points in k-dimensional space but are distributed in a restricted manner that either allows an easier search for the nearest neighbor or else allows for a search that does not necessarily produce the nearest neighbor. In this section, seven different constrained structures are described, all of which have been used as part of subband/VQ schemes. The imposed structural constraints return computation savings, memory savings, or sometimes both. Due to space constraints, we limit ourselves to describing the functioning of each structure, but references are provided for design algorithms.

1) Lattice VQ: A lattice Λ_n in $\mathcal{R}_n$ is composed of all integral combinations of a set of linearly independent vectors that span the space. In one dimension, a lattice quantizer is a uniform quantizer with a (theoretically) infinite number of levels, where the entire real line is partitioned into a countable set of equal size intervals of length Δ. In 2-D, a lattice is a set of points with a regular arrangement in the plane:

$$\mathcal{L} = \{\mathbf{t} \in \mathbf{R}^2 : \mathbf{t} = \mathbf{A}\mathbf{m} \text{ for all integer pairs } \mathbf{m} = (m_1, m_2)\}$$

where $\mathbf{A}$ is a nonsingular (2×2) matrix called the generator matrix of the lattice, and $\mathbf{m}$ is an integer vector. This concept generalizes to N dimensions. Conway and Sloane [53], [54] have determined the best known lattices for several dimensions, as well as fast quantizing and decoding algorithms. Some important n-dimensional lattices are the root lattices $A_n (n \geq 1)$, $D_n (n \geq 2)$, and $E_n (n = 6, 7, 8)$, the Barnes–Wall lattice Λ_{16} in dimension 16, and the Leech lattice Λ_{24} in 24 dimensions. These latter give the best sphere packings and coverings in their respective dimensions. Two critically important issues in lattice VQ concern the truncation and scaling of the lattice. The truncation region, which is also called the region of support, is the subset of the lattice that will actually be coded. For those input points in the "overload region," that is, the region of possible input space that extends beyond the chosen lattice subset, the output distortion ("overload distortion") is not bounded in the same way as it is in the interior of the lattice ("granular distortion"). For each cell in the region of support of a lattice VQ, the reproduction vector is taken to be either the midpoint of the cell or else the centroid of the training vectors lying in that cell. It is desirable to scale the lattice (or equivalently, the source) to minimize the total distortion. The choice of support region can also have an important effect on the encoding speed.

2) Gain/Shape VQ: When the distortion measure is squared error, gain/shape VQ provides a simple means of increasing vector size and sequentially coding the vector shape and its gain without requiring any normalization of the input vector. Input vectors are first encoded into a unit energy shape vector by a maximum correlation search, and then the optimum gain for the given shape is found by a simple SQ. The product codebook is suboptimal in theory because it imposes a structural constraint on the codebook, but in practice it can improve performance by allowing larger dimensions for the shape codebook [55]. When a product codebook is used, the sequential encoding rule is optimal in the sense of minimizing MSE [55].

3) Tree-Structured VQ (TSVQ): In TSVQ, the codeword is selected by a sequence of binary minimum distortion decisions comparing the input vector to stored vector reproductions at each available node. The encoder produces binary symbols to represent its sequence of decisions from the root node of the tree through the terminal node. This binary sequence or pathmap is the final codeword. Code trees can be *balanced* if all indexes have the same length R, yielding a fixed rate code, or *unbalanced* with indexes of differing length, yielding a variable rate code. Compared with a full-search unstructured VQ, the search complexity of a balanced tree is linear in the bit rate instead of exponential but at the cost of a roughly doubled memory size. For unbalanced trees, the search complexity remains linear in the *average* bit rate, but the memory can be considerably larger unless constrained [41].

4) Multistage VQ: Multistage VQ divides the encoding task into several stages [44], [56]–[58]. The first stage performs a relatively crude encoding of the input vector using a small codebook. Then, a second-stage quantizer operates on the error vector between the original vector and the quantized first stage output. The quantized error vector provides a refinement to the first approximation. At the decoder, the reproduction vectors produced by the first and second stages will be added together. Additional stages of quantizers can provide further refinements. Unlike full-search VQ, whose encoding complexity and memory requirements increase exponentially with the dimension-rate product, in multistage VQ, the increase is only linear. This has particular utility in subband coding since either the rate or the dimension can be made large, which allows it to respond to the occasional need for a locally high bit rate in subband coding.

Multistage VQ is sometimes referred to as residual VQ, but those words better apply to a variation on multistage VQ that effectively uses the same codebook but allows a nongreedy search such as a full search or an M-algorithm. The codebook corresponding to a multistage VQ can be viewed as the collection of all possible reproductions that can be constructed by adding one codeword from each of the stages. This codebook can be viewed as a direct sum codebook because it corresponds to all possible sums of one reproduction codeword per layer or as a product codebook because the Cartesian product of the indexes provide a composite index that can be used to produce the final reproduction. If this codebook is searched in a greedy fashion one stage at a time, the resulting code is naturally progressive, but it need not find the best of all possible reproduction words in the direct sum codebook. Allowing the encoder to look at the entire reproduction will lead to better performance, but the search complexity grows accordingly. Design algorithms specific to this encoder structure can then be used [59]–[61].

5) Predictive VQ (PVQ): PVQ is a straightforward vector extension of traditional scalar predictive quantization (DPCM). The encoder makes a prediction of the incoming vector based on previously encoded vectors. The difference between the actual input vector and its prediction is called the residual vector. This residual is vector quantized. Because the encoder only uses the previous *outputs* in making its prediction, the decoder is able to make the same prediction. After dequantizing the residual vector, the decoder adds the prediction to it to form the reproduction vector. The prediction is often a simple linear predictor that takes a weighted average of nearby previously encoded coefficients.

6) Classified VQ (CVQ): The input x_n is subjected to a classifier that generates an index i_n, an integer from 1 to m, and this identifies which subcodebook C_i to search for the nearest neighbor [62]. The codeword corresponding to x_n thus consists of two parts: an index i_n specifying the codebook and an index u_n specifying the selected word in that codebook. The first part of the codeword is typically referred to as *side information* although it can be longer or shorter than the second part. The classifier can be based on texture properties, edge directions, features of diagnostic or scientific importance or irrelevance, perceptual masking, or a variety of other criteria.

7) Finite-State VQ (FSVQ): Like a classified VQ, a finite-state VQ has multiple codebooks, and the next vector to be encoded is referred to the appropriate codebook. Unlike CVQ, however, no side information is sent to the decoder to inform it of which codebook to use. That information is inferred using the *next-state* rule. The next-state rule uses various information available to the decoder in order to decide which codebook to use. For instance, reconstructed vectors from nearby locations, the codebook that was used the last time (corresponding to the previous state of the system) and the codeword that was chosen from that last codebook, are all pieces of information available to both the decoder and the encoder that can be used to decide which codebook to use for the current vector (corresponding to the current state of the system). The decoder and encoder both know the next-state function. Assuming that the decoder and encoder start with the same codebook and that there is no channel noise, the decoder will always be able to track which codebook the encoder is using next since the next-state function employs only information available to both parties [63], [64].

8) Trellis-Coded VQ (TCVQ): Trellis-coded quantization (TCQ) and TCVQ can be viewed as an extension of the FSVQ approach with a lookahead search. They are source coding analogs to the Ungerboeck coded modulation approach [65] to channel coding (error control coding) on narrowband channels. The TCVQ encoder can be thought of as having a supercodebook that is partitioned into a collection of subcodebooks. At any time, the encoder may have only a few of the subcodebooks available. A popular choice is to have the supercodebook be a lattice and the subcodebooks be sublattices. The encoder has a collection of possible states and allowed transitions between those states. Each of the transitions between states corresponds to one of the subcodebooks. As a form of FSVQ, this system's encoder in a given state could view a single source vector, look at all of the available subcodebooks, and find the nearest neighbor. The encoder could communicate this word to the decoder with a two-part binary vector. The first part tells the encoder's next state by sending an index of which allowable transition is being taken. The second part indexes which of the possible codewords in that particular codebook was chosen. This implementation, however, suffers the shortcoming of allowing a good short-term choice to lead to a bad state, which could in turn lead to bad long term behavior. Hence, the TCVQ is allowed to look ahead to find the best available sequence of paths for a sequence of input vectors. This ensures better long run behavior at the expense of added coding delay. A variety of design algorithms for the supercodebook and subcodebooks based on lattices and on clustering have been considered [66]–[68].

9) Hierarchical VQ (HVQ): Although the adjective "hierarchical" has been applied to many variants of VQ (and other things), we here consider the technique of Chang *et al.* [69] for replacing nearest neighbor computations by a fast table lookup. We illustrate the idea for an 8-D vector. Suppose that we have a vector $(Y_0, Y_1, \ldots, Y_7)$ with 8 b/symbol. We wish to construct a code of 1 b/symbol. Using a full-search codebook would require computing the squared error between this input vector and all 2^8 vectors in the codebook. A TSVQ would reduce the computation to eight binary comparisons. HVQ can reduce this to no computation at all. Off-line, each of the four input pairs (Y_0, Y_1), (Y_2, Y_3), (Y_4, Y_5), (Y_6, Y_7) is encoded using (possibly different) 8-b 2-D codebooks, that is, codebooks having a total of 2^8 words. Subsequently, one can implement that mapping of input pairs into codewords via table lookup. There are $2^8 \times 2^8 = 2^{16} = 65536$ possible input pairs and $2^8 = 256$ possible codewords; therefore, a table providing one of 2^8 indexes for each 2^{16} possible input pairs provides a nearest neighbor mapping from all possible inputs to all possible reproductions. Furthermore, this stage achieves a compression of 2:1 since the total of 16 b for the original vector (2-D vector with 8 b for each coordinate) is now reduced to 8. The next stage takes the reproduction pairs and combines them two at a time to form two 4-D vectors. These are each quantized with a (possibly distinct) 8-b codebook, where a table lookup again replaces the precomputed nearest neighbor mapping. The process can be repeated once more. The overall operation results in a compressed vector at 1 b/symbol with no computation. The HVQ has the nice property of being easily scalable, as one can put out indexes at rates of 8, 4, 2, or 1 b/symbol. Further stages will yield further compression. The overall operation is depicted in Fig. 6.

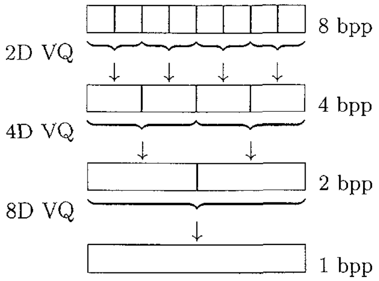


Fig. 6. Hierarchical VQ: Each arrow is implemented as a table lookup having 65 536 possible input indexes and 256 possible output indexes. The table is populated by an off-line minimum distortion search.

The obvious cost of this method is memory, but this can be an acceptable price to avoid numerical computation. The method can be applied to an image by sequentially compressing rows and then columns. Hierarchical VQ did not have much impact in its original form because it was essentially a means of implementing simple VQ on raw data, e.g., speech samples or image pixels, and such simple VQ, even if it is full search, does not usually perform well in comparison with transform or subband codes. However, HVQ can perform both subband coding and VQ together by table lookup, resulting in a fast software-only implementation of subband/VQ image coding systems [70].

IV. Vector Quantization for Image Subbands

In this section, we present a survey of many algorithms that use VQ on the coefficients from a subband decomposition. The algorithms are grouped initially according to whether the VQ operates in a purely intraband mode or whether it employs any interband information. One could argue that there is no such thing as a purely intraband subband coder, that is, one that quantizes the bands entirely separately. At the least, the coder must make some decision about how to allocate bits among the bands, and the quantization process cannot be said to be completely separate if the bands share a fixed total number of bits. We will refer to this situation, however, as an intraband coder and use the terms "interband VQ" or "crossband VQ" to indicate any algorithm that attempts to exploit explicitly the correlation between the bands. Such methods include those systems that quantize coefficients from different bands in common vectors or that explicitly use information from one band to affect the coding in another band.

Intraband methods have used a wide variety of different VQ structures, and Section IV-A on intraband methods is organized according to the VQ structure. Section IV-B includes the relatively few methods that form the vectors intraband but use interband information to better code the intraband vectors. The bit allocation optimization in this situation is more complex.

Section IV-C discusses those methods that form the vectors across bands. This can be done in a number of different ways, and Section IV-C is organized according to vector-forming strategies. The bit allocation problem becomes quite different in these cases since there may be no explicit assignment of bits to bands. When coefficients from different bands form a vector, the target rate of the VQ does not simply and directly allocate bits to each band, as with intraband VQ's. The use of a training sequence can indirectly allow greater rate to those bands with greater variance; weighted distortion measures and other means can also accomplish this. Section IV-D discusses subband/VQ systems for color images. Such systems often form the vectors out of coefficients from different color planes.

Most researchers chose to test their algorithms on the 8-b grayscale image called Lena of size 512×512 and to report their results as peak-signal-to-noise ratio (PSNR), which is defined by

$$\mathrm{PSNR} = 10\log_{10}\frac{255^2}{D} \tag{1}$$

where D is the MSE between the original and compressed test images. When specific PSNR results are reported for this image, we include a brief summary of those values. There are other images that are considered to provide a more rigorous test of a compression system, and within the information theory community, researchers often use simulations of ideal sources (e.g., performance of a quantizer for a simulated unit-variance Gaussian source). However, the results reported for Lena have some utility for comparison purposes. When the Lena image of size 256×256 was used, we included those results, with the knowledge that it is more difficult to achieve equally high PSNR's on that smaller image. In this paper, when the size is not stated, the larger one is understood. In some cases where other test images were used, we briefly mention the reported results, although for results on color images and video sequences, refer to the specific papers in question. We note that when an algorithm attempts to minimize a perceptual distortion measure rather than MSE, the PSNR results are of little use and are generally not reported.

A. VQ on Separate Bands

When quantizing the subbands separately, the quantizer design problem involves the selection of a vector dimension (and vector shape) in each subband as well as a bit rate for each subband. This bit allocation is often achieved in an ad hoc way. Some methods solve an optimization problem in which the overall quantization distortion is minimized subject to a constraint on the overall bit rate. The resulting quantizers can be unrealizable due to excessive complexity or memory requirements, and therefore, the optimization methods usually invoke constraints on encoding complexity or memory as well. Additional sophistication can be added to the optimization problem by including perceptual factors in the distortion measure or by having the optimization choose the vector dimension for each subband as well as the bit rate. With simple quantization schemes like PCM, Lagrange multiplier solutions to the optimal bit allocation problem exist. For more complex subband VQ strategies, it can be solved by conventional nonlinear smooth optimization techniques.

The vector-forming strategy is usually simple in the intraband case. Most systems employ square blocks of size 1×1, 2×2, or 4×4, together with the obvious tiling strategy. Some systems used vectors of size 2×4 or 4×2, which provide an intermediate dimension with a still-simple tiling strategy. Such rectangular blocks are usually taken to be oriented along the direction of the detail (either horizontal or vertical) of each high frequency subband, because that way, there is greater intercoefficient correlation. One study that examined codeword orientation concluded that these rectangular vectors oriented along the direction of greater intercoefficient correlation were superior to vectors oriented in the perpendicular direction [71]. Two other studies suggested that under particular quantization strategies, there might be advantages to scrambling the coefficients within a band when forming intraband vectors [72], [73].

1) Multistage and Residual Vector Quantizers: The earliest effort at separately encoding the bands focused on multistage VQ [74]. Early work was with a two-stage coder [74], [75], and a later study used a variable-rate coder with as many as 48 stages [76].

In [74], a two-band partition in horizontal and vertical frequencies yielded a total of four subbands, the lowest of which was encoded using multistage VQ. The error in the first stage was quantized to refine the reconstruction. The bit allocation algorithm used the fact that perceptually significant errors tend to occur in clusters in high-entropy parts of the input image. Application of this technique to a 512×512 monochrome image resulted in good quality at 0.5 b/pixel. Higher quality at the same bit rate was attained by increasing the number of subbands to 16.

Multistage VQ for subband coding of super high-definition images was developed in a series of articles [77], [75], [78]. Several QMF's were examined from the viewpoints of reconstruction accuracy, coding gain, and low-pass characteristics. A two-stage encoder was introduced in which the first stage is applied to the subband coefficients, and the second stage is applied in the *spatial* domain. The system employed 4×4 blocks in a four-band decomposition. The lowest band was scalar quantized. Bit allocation was proportional to the logarithm of the subband variance. Codebooks were designed by the GLA.

Variable rate multistage VQ was used in [76]. Variable rate multistage VQ examines the energy of the residual vector after each stage. If it falls below a predetermined threshold, the encoding stops. The method requires side information to inform the decoder of the number of stages to expect for each vector. The vectors were first classified according to their energy level, which was scalar quantized separately with a Lloyd–Max quantizer. Each vector was then quantized with either a fixed- or variable-rate multistage VQ, depending on the vector size. In the different levels, 4×4, 8×8, and 16×16 vectors were used. For each stage, the classified VQ's used two or four classes. Each class contained four code vectors. No theoretical analysis was performed to allocate bits. The decomposition used a 20-tap Daubechies wavelet filter. Results on Lena were 30.06 dB at 0.222 b/pixel, with reasonable complexity and memory requirements. The lowband attained a PSNR of 41.14 dB at 6 b/pixel using a fixed-rate 48-stage VQ with four code vectors per stage operating on 16-D vectors. This effort is in contrast with almost every other study surveyed in that most work uses SQ or tiny vector sizes for the low band.

One study used multistage VQ with a multipath search of the direct sum codebook, which could almost yield the performance of a full-search while having only slightly more computational complexity than a sequential search [79]. Because it is an iterative descent algorithm based on a Lagrangian minimization, the entropy-constrained residual VQ (EC-RVQ) design algorithm attempted to jointly optimize each stage codebook to minimize the reconstruction error over all the training data subject to a constraint on the output entropy of the RVQ. The system could be used with large vector sizes (e.g., 8 × 8). The decomposition was into seven bands using a computationally efficient IIR allpass polyphase filter bank based on 2-D decompositions. Bit allocation was by the "equal-slope" method. For the Lena test image, PSNR's of 28.28 and 31.05 were achieved at 0.10 and 0.16 b/pixel, respectively.

Semi-orthogonal spline-wavelet-packets [80] were used with interpolative/residual VQ on the baseband and full-search VQ on the other bands in [81]. Before quantization, a decision was made for each subimage of the percentage of coefficients to be retained, which depended on the desired compression ratio. Those wavelet packet coefficients with magnitude below a threshold would be set to zero. If the number of nonzero entries in a vector was below a predetermined value, the vector was set to zero. Compression results were PSNR's of 29.02 and 27.47 dB at 0.1 and 0.08 b/pixel, respectively, for the Lena test image.

Multistage subband VQ was shown to be robust against fading channels in wireless communications [82]. The idea is that the burst error behavior typical of fading channels can cause the loss of several bits describing a single stage in a multistage VQ. Provided the lost stage is not the first stage, the remaining stages will still provide a good overall reproduction. The first stage bits are given the extra protection of error correcting codes. The particular implementation uses a seven-band decomposition with Johnston QMF filters, a three-stage VQ for the baseband, and scalar-quantized individual coefficients with entropy-coded location information for the higher bands. Performance was reported on a simulated digital european cordless telecommunication (DECT) system to be 35 dB at 0.69 b/pixel and a channel SNR of 15 dB.

2) Lattice Vector Quantizers and ECVQ: For the encoding of intraband subband coefficients, lattice VQ has been explored more than any other single type of VQ. One series of articles used a biorthogonal wavelet decomposition in the form of a quincunx pyramid together with a lattice quantizer based on a suitably scaled and truncated version of the Λ_{16} lattice [83]–[85]. For Lena, a PSNR of 30 dB was obtained when the entropy rate was 0.064 b/pixel. However, the actual bit rate obtainable for this PSNR was not specified. Their work compared the quincunx pyramid and the Λ_{16} lattice against the dyadic wavelet and the E_8 and D_4 lattices, finding the former combination to be superior. The later work focused on the issues of scaling and truncating the lattices and counting the number of points within the truncated area [85]. The basic method was to truncate the lattice as a pyramid, encoding those input vectors inside the truncation region by finding the nearest lattice point and encoding those input vectors in the overload region (outside the truncation region) by projecting onto the pyramid's outer shell. The index for the nearest lattice point on that outer shell would then be encoded, as well as a quantized version of the projection distance, so that the decoder could reconstruct those vectors by projecting back out. Results were a PSNR of 30.3 at 0.174 b/pixel for the Lena test image. A related paper examined the use of lattice VQ with an elliptical truncation region [86]. The elliptical region of support was thought to be useful because neighboring wavelet coefficients retain some positive correlation; in two other studies, this perceived problem was solved by removing the correlation by forming vectors from nonneighboring coefficients [72], [73].

Another study pointed out that the entropy rate achieved in [84] was not at all representative of the actual bit rate achievable because the number of available codewords in the lattice codebook could be orders of magnitude larger than the number of codewords actually used by a given test image [87]. This work also examined a drawback to the pyramidal truncation shape of [85]. Input vectors lying outside the truncation region could be in a "columnar region" or a type of "wedge region," depending on whether they project to a lattice point with all nonzero coordinates or to a lattice point with one or more zero coordinates. If the projection distance were not quantized, the distortion incurred by quantizing input vectors falling in a columnar portion of the overload region would be bounded by the shape of the Voronoi region of the lattice points in an untruncated lattice. However, the distortion incurred by input vectors in a wedge region of the overload zone might not be bounded except by the bounds on the input vector range itself. This drawback was discussed and solved in [87] by using a dual density lattice composed of two concentric pyramidal volumes of lattice points both centered on the origin with the inner pyramidal volume having a denser placement of lattice points. By allowing the density of lattice points in the outer volume to be much lower, the truncation of the outer lattice did not need to be so severe, and the volume of the overload wedge regions could be reduced as much as desired. The design algorithm then involved selecting scale factors and pyramidal heights for both lattice sections.

Lattice VQ was compared against ECVQ and against unstructured full-search VQ designed by the GLA in [43]. A four-level octave-band decomposition with nine-tap QMF's produced 13 subbands. Bit allocation used the "equal-slope" method. Initial comparisons were made for 8-D vectors from selected subbands. ECVQ performed best at low rates. The E_8 lattice did best at higher rates, possibly because ECVQ was strongly constrained by its limited number of codewords. The Z_8 lattice with centroid reproductions did almost as well as E_8. Most of the lattice VQ gain over SQ was found to be due to the joint encoding and joint centroids; the denser sphere packing of E_8 seemed less important. Using E_8 for the finest scale, D_4

for the next scale, and A_2 for the coarsest levels resulted in a PSNR of 30.9 dB at 0.136 b/pixel (where that cited rate was an entropy rate and assumed an ideal entropy coder). The authors mentioned the utility of shifting the subimage by one pixel horizontally, vertically, or diagonally prior to decomposing and downsampling it further as a means of extending the training sequence size at the coarser levels.

A method for incorporating some features of human vision within the framework of subband lattice VQ was presented in [72]. A 13-band decomposition was performed with a Daubechies wavelet filter. Vectors of four and 16 dimensions were formed intraband. Vectors composed of distant coefficients were found to provide better results than vectors composed of adjacent coefficients, although details on the vector composition were not provided. Psychovisual experiments were performed to determine the maximum unit size for each lattice and for each subband such that no granular noise would be visible. This threshold, which was called the differential visibility threshold, depended on the viewing distance and on masking effects and was used (approximated by a piecewise linear function) to normalize each input vector prior to quantization. The Z lattices were found to be generally superior to the $D, E,$ and Λ lattices. Transparent quality on high-quality monitors was achieved at between 0.42 and 0.78 b/pixel, depending on the complexity of the input image.

A comparison of the granular distortion and overload distortion for various lattices on subband coefficients was presented in [73]. For generalized Gaussian sources at low bit rates, the Z lattice was shown to provide better performance and lower computational complexity than the E_8 and Leech lattices.

A small number of studies have examined the flexibility of the wavelet packet decomposition [15] for image coding [88]–[90]. One study that used VQ employed nonstationary wavelet packets based on filters varying at each stage of the tree-structured decomposition [90]. The complete tree was pruned into the best basis subtree which minimized a perceptually weighted distortion [91], [92] for a given bit rate. Each subband was separately quantized with lattice VQ's (D_n and E_n) with a dead zone. Bit allocation was by the "equal-slope" method.

A trellis-coded VQ based on lattice codebooks was used in an intraband fashion on the decomposition produced by separable biorthogonal wavelet filters of nine and seven taps [93]. In the different subbands, the VQ dimensions were 1, 4, 16, and 64, and the bit allocation was based on [94]. Results on Lena were 28.9 dB at 0.265 b/pixel and 29.2 dB at 0.281 b/pixel.

One study compared lattice VQ and kd-tree structured VQ for scalable video compression [95]. The goal was to decouple the frame rate, resolution, and bandwidth of the encoding from those of the decoding by using successively larger codebooks that were nested subsets of the largest codebook.

A hybrid SQ/ECVQ system was used in [96]. The input image was decomposed into 10 octave-band subbands by three stages of a 12-tap QMF. Subbands were blocked into 4-D vectors. If the standard deviation of an input vector was below a threshold, it was quantized to zero. Otherwise, the component magnitudes were individually compared against a different threshold. If they were all below it, the vector was quantized with an ECVQ. If not, the components were scalar quantized with a modified scalar quantizer that had uniformly spaced decision thresholds but output levels based on interval centroids. In either case, the output indexes were Huffman coded. Use of the VQ provided an improvement in both SNR and subjective quality compared with SQ with run-length entropy coding or with block Huffman coding.

3) Pruned Tree-Structured Vector Quantizers (PTSVQ): PTSVQ and lattice VQ used for packet video were both examined in [97]. One scheme used an octave-band decomposition with the 16C QMF filter, resulting in seven subbands. Separate PTSVQ's were designed by the GLA for each subband. The tree structure can be exploited in a straightforward manner for layered ATM video coding. High-priority packets contain the initial information from a smaller subtree, and enhancement-layer packets, subject to cell loss, code deeper into the tree. A second scheme used time-domain aliasing cancellation parallel filter banks to produce 64 uniform subbands. These were lattice quantized (Λ_{16}, E_8). Samples of magnitude below a given threshold were excluded from VQ encoding via run-length coding. The remaining samples were grouped together for VQ coding. The method is not quite a pure intraband method; if a vector could not be completely filled with the samples from one subband, samples from the following subband were added. The enhancement-layer coder was used for the residual error from the base-layer coder. The algorithm involving 64 uniform subbands with lattice VQ was found to be superior to the seven-band octave band decomposition with PTSVQ; therefore, the former algorithm was investigated with a variety of cell-loss rates, motion-compensated prediction, and interpolation schemes.

TSVQ was also used intraband in [98] and [99]. In the former work, 2×2 or 4×4 blocks were coded from a four-band decomposition of image sequence data. Motion compensation was conducted on the low band. A small subtree was used for encoding unless the MSE of the quantized block exceeded a predetermined threshold, in which case, a large tree would be used. In [99], binary trees were created for 4-D vectors for the higher six bands of a 10-band pyramidal decomposition using a five- to seven-tap biorthogonal wavelet filter. Blocks were of size 4×1, 1×4, or 2×2 to follow the detail direction. Given the large starting size of the bands for these super high-definition images, large superblocks (of size 16×16) that could be zeroed out all at once for greater efficiency were defined.

4) Full-Search Unstructured VQ's: Full search of an unstructured codebook has been examined by many researchers. Separate VQ's were designed by the GLA for each resolution level and preferential direction in [100]–[102], [94]. The first of these used a recursive biorthogonal wavelet transform and was basically a comparison of linear and modified linear spline filters. The second was a comparison of the GLA and the Kohonen neural network (KNN) or self-organizing feature map design algorithm. The KNN method uses a neural clustering network in which the weights, or VQ codewords, between a large number of locally interconnected nodes adaptively self-organize their values in order to reflect the best feature maps, or codebook, representing the input vector patterns. For the subbands of the recursive biorthogonal wavelet transformed image, the unstructured full-search codebook produced by the KNN algorithm was found to yield approximately the same PSNR as one produced by the GLA, using about 15 times less computation during training [101]. The latter two papers used a five-stage biorthogonal wavelet decomposition and encoded with scalars in the lowest band, 4-D vectors in medium bands and 16-D vectors in the high bands. In this work, *noise shaping* was applied across the subbands by minimizing a weighted MSE where the weights were proportional to the log of the variance of each subband, and the constants of proportionality were chosen experimentally to match human vision. The distortion expression was minimized subject to the total bit allocation constraint with a Lagrange multiplier method. The authors included a comparison of different wavelet filters.

Another use of visual masking with subband VQ used a uniform 16-band decomposition with a masking function associated with each coefficient [103] in a manner similar to [36]. The distortion measure was designed to ignore differences between elements of the two vectors that are smaller than the threshold suggested by a visual masking function. The lowest band was coded with DPCM, and higher subbands were coded with full-search unstructured VQ. The bit allocation was determined by an optimization procedure that minimized a distortion expression subject to constraints on the total bit rate, the complexity (maximum codebook size), and the possible vector sizes.

Nonseparable filter banks with hexagonal regions of support were applied to hexagonally sampled (or resampled) images in [104]. The resulting subbands had hexagonal shapes, with higher subbands having twice the bandwidth of lower ones. The subbands then had orientation properties which were not limited to vertical and horizontal directions, and the basic structure is believed to be similar to that found in the human visual system. (In Section IV-D, an algorithm is described that used diamond-shaped filter banks [105].) Full search unstructured VQ's were designed for each subband separately using the GLA. Performance results were not given.

Another study that used hexagonal filter banks with unstructured full-search VQ was [106]. Vectors were of size 4×4 for all bands of a 10-band decomposition. Variable sized multiresolution motion estimation was performed. Bit allocation was based on the log of the variance and on the determinant of the covariance matrices, together with a weighting factor that accounted for some simple aspects of a visual model.

Although not precisely "image" data, the output of supercomputer ocean models, consisting of values of six different physical variables at each point of a large rectangular 3-D grid, represents an interesting application of wavelet/VQ methods [107]. Using a symmetric biorthogonal wavelet filter on 2-D data [108], four levels of octave-band decomposition resulted in 13 subbands. Full-search unstructured VQ's were designed for each subband using training data obtained from the first 91 time steps in the sequence. The optimal codebook sizes and vector dimensions for each combination of total bit rate and total complexity were obtained by a nonlinear optimization procedure as in [109]. The experiment was also repeated with first-order frame prediction (in which the prediction for the next frame was simply the quantized version of the previous frame), which provided 3-5 dB gain over the nonpredictive version.

5) Classified VQ's: An intraband CVQ was used on the motion-compensated frame difference signal decomposed into four subbands by an eight-tap separable QMF [110]. The fourth band was discarded. The remaining bands were divided into 4×4 vectors. Blocks in subbands 1 and 2 were assigned to one of four classes based on the variance of the coefficients in the block. The lowest variance class was not encoded, and the other three used three different VQ's. The third subband used only two classes of which the lower variance one was not coded. The VQ's were all full-search unstructured VQ's designed by the GLA.

In [111], each image in the sequence was decomposed into seven bands with the Daubechies four-tap filter. Blocks of size 4×4 were used for the first pyramid level and 2×2 blocks for the second level. The first frame of a sequence was used to generate full-search codebooks for each subband. In all subsequent frames, each vector was compared with the corresponding vector in the previous frame. The difference was classified by being compared with some small threshold, and side information informed the decoder of the result. If the difference was less than the threshold, no further action was taken for that vector. Otherwise, the vector was quantized and used to update the cluster mean. If the difference between the current codewords and the previous ones exceeded a predetermined threshold, new codewords were transmitted to the decoder. With smaller codebooks for the diagonal directions, a PSNR of 36 dB was achieved at 0.35 b/pixel for the first 20 frames of the "Claire" sequence.

A combination of classification with PTSVQ and multistage VQ was proposed in [112] as being particularly well suited to the design of scalable compression systems.

6) Other VQ Structures: An early intraband method used alphabet and entropy constrained VQ (AECVQ) [113], which was designed as follows. An initial large codebook was designed by the GLA for each subband. Codeword lengths were assigned to codewords based on their probabilities. Successively smaller codebooks were chosen from these same codewords by an iterative process. Each training vector was mapped to the codeword which minimized a weighted combination of MSE and codeword length. Then, the centroid of each partition cell was computed, and one of the preexisting codewords was reassigned to that cell. The reassignment alters both the overall rate and distortion of the code. In a manner similar to PTSVQ, this AECVQ allows one codebook to provide a range of distortion/rate operating points. A three-level octave-band decomposition yielded 10 subbands. Vectors of size 2 were used in the lowest four bands, and 4-D vectors were used in the others. Bits were allocated among the bands by the "equal-slope" method. For Lena, PSNR was 34.3 dB at 2 to 3 b/pixel.

One intraband method used a "linear" VQ codebook, which is a structure unlike any of the ones described previously [114]. The image was split into seven subbands using separable QMF's (eighth-order Johnston's filter at the first stage and lower orders at the next stage). Lloyd–Max SQ was used on the baseband, and VQ was applied to the other bands in dimensions of 4, 8, or 16. Bits were allocated based on the variances of each subband, which had to be transmitted to the decoder in order to recompute the allocation there. For a vector of length N, the "linear" codebook consists of M concatenated symbols, where $M >> N$. To find the appropriate codeword for an input vector, the input vector of length N is first compared against the first N symbols of the linear codebook. Then, it is shifted over by one and compared against symbols $2, 3, \cdots, N+1$. For a given number of possible reproduction vectors, the linear codebook takes up much less space than the full-search unstructured codebook. This method was used with prediction and with adaptive arithmetic coding applied to the output bit stream. For the Lena test image, the authors claimed transparent quality at 0.2 b/pixel.

In [115], the pyramidal structure of the wavelet decomposition was used to advantage with a hierarchical motion compensation technique, which significantly reduced the computation of motion compensation. A second-order separable QMF was used to produce seven octave-band subbands. Motion estimation performed on the baseband was refined using the baseband of the higher level and using the original image. The frame difference image was encoded using a fast search algorithm known as approximation elimination search [116].

Gain/shape VQ was used with variable-sized rectangular intraband vectors in [117]. A prediction error image from a motion-compensated image sequence was decomposed into 16 subbands by separable QMF's. Each subband was divided into regions that would get encoded and ones that would not. For each region to be encoded, the encoder would have to describe to the decoder the location of the region in addition to encoding the coefficients for the region. Choosing the regions to be square blocks would cost a few bits for location information but would include many irrelevant (low magnitude) coefficients. Arbitrarily shaped regions would consume almost no data rate for the encoding of irrelevant coefficients but would cost much overhead for location information. The compromise chosen was to allow nine different rectangular shapes up to a maximum size of 3×3. For each candidate group of rectangles, a quality measure was based on the visual weighted sum of prediction error energy covered by the selected rectangles. The set was iteratively improved by a greedy relaxation algorithm that could add or remove rows or columns to the rectangles within the data rate constraint. The chosen rectangles were encoded with a gain/shape VQ of the appropriate dimension designed by the GLA. The storage requirements for the nine VQ's could be constrained by mapping all codebooks of smaller dimension into the codebook of the largest dimension.

7) Additional Transforms: A few studies examined the use of additional transformations within the subband/VQ framework. For instance, one can use a DCT on the baseband of the decomposition and then scalar quantize the DCT coefficients. Only two algorithms used additional transforming on all (or most) of the bands following a subband decomposition and preceding VQ.

A wavelet/VQ algorithm incorporating a neural network was presented in [118]. Neural networks can compress data by approximating a data vector with a vector of reduced dimensionality. A QMF was used to decompose an image into seven octave-band subbands. The highest of these was not coded, and the lowest was coded with DPCM. The neural network structure consisting of n units at the input and output layers and k units in a hidden layer (where $k < n$) was applied to the remaining bands. The k-dimensional vectors of the hidden layer were quantized by full search of an unstructured VQ designed by the GLA. A PSNR of 29.1 dB was obtained at 0.25 b/pixel for the 256×256 Lena test image.

Vector transform coding (VTC) is a vector generalization of conventional transform coding techniques such as JPEG. VTC includes a vector transform (VT), which decorrelates vectors, followed by VQ [119]. The vector transformation does not reduce the intervector correlation as much as the DCT does, but it preserves the intravector correlation much better, which makes subsequent VQ more efficient. A subband/VT/VQ scheme was proposed in [120]. The image was decomposed into seven subbands with the Daubechies four-tap filter. Each band was independently transformed with the 2-D VT. Bits were allocated among subbands in a heuristic way and were allocated among vectors within a band in a way that considered both perceptual importance and energy distribution. The method provided a PSNR of 29.2 dB for Lena at 0.25 b/pixel.

B. Interband Information for Intraband Vectors

The correlation between coefficients in different subbands can be exploited in a variety of ways. Section IV-C treats those algorithms that form vectors across bands. This section looks at ways in which crossband information can be used to advantage in the coding of intraband vectors. For example, the zerotree structure can be used to predict insignificance of coefficients in one subband using knowledge of coefficients in another. Finite-state VQ can be used by forming vectors intraband and using crossband information to determine the state. In this section, we survey these various methods.

Pentland and Horowitz [121] used the PNN algorithm to design separate VQ's for each subband in a standard QMF pyramid but then estimated conditional probabilities across subbands for the VQ outputs by simple conditional histograms. These crossband conditional probabilities were then used in a conditional entropy coding scheme using both Huffman and run-length coding. This provided the quality of the original VQ scheme at a much-reduced bit rate. An example reported 34.9 PSNR at 0.42 b/pixel.

Podilchuk *et al.* presented a system in which vectors were composed of intraband coefficients, but interband information was used to obtain a localized codebook search strategy [122]–[124]. Ten-tap 1-D QMF's were first used temporally, then in the spatial horizontal direction, and then in the spatial vertical direction, resulting in eight bands. Another round of spatial filtering split the low band into 4. Of the 11 bands, only these four and the low-spatial/high-temporal frequency band were coded. In [122], the lowest band was coded with SQ. Noting the sparse nature of the higher bands, a VQ codebook was designed with 960 "sparse" binary vectors of size 9 × 9. Each codevector consisted of either a single strip or a single dot of various widths and sizes. By transmitting the index number for one of the codebook entries along with the foreground and background intensities to be used as the two binary levels, the upper subbands could be coded at no greater than 0.3 b/pixel. This method was called geometric VQ (GVQ). The encoded low spatio-temporal frequency band was used to reduce the codebook search and estimate the local areas in the remaining bands where perceptually significant edge-like data is located. Using the low band to estimate the structure of the higher bands required no bits and reduced the data rate and search complexity. In later work [123], [124], delay was reduced by substituting a two-tap Haar filter for the temporal decomposition, the GVQ codebook was designed for 3 × 3 vectors, and different coding strategies for the lowest band were examined (adaptive DPCM and unbalanced TSVQ).

A few studies attempted to predict insignificance across the bands using the zerotree idea developed by Shapiro [28]. In an octave-band decomposition, each coefficient X_i (except those in the lowest band and the three highest bands) is related to exactly four coefficients in the next higher band. Those four coefficients correspond to the same orientation and spatial location as X_i does in the original image. Each of these four is in turn related to four in the next band, and so on. These coefficients are collectively called the *descendants* of X_i. The relationship is depicted in Fig. 8. In an octave-band decomposition, it is often true of image data that when a coefficient X_i has magnitude less than some threshold T, all of its descendants will as well. The collection of coefficients is then called a *zerotree* with respect to the threshold T, and the coefficient X_i is called the *zerotree root*. Shapiro's embedded zerotree wavelet (EZW) algorithm uses this zerotree structure to efficiently "divide and conquer" the coefficients in an iterative SQ approach. The zerotree root is the wavelet equivalent of the end of block (EOB) symbol in JPEG coding. That is, when a sequence of scanned coefficients ends with a tail of zeros, one simply cuts off the tail with a special symbol. The zerotree grows exponentially with the depth, whereas in JPEG, there is no such growth, thus making the zerotree potentially more powerful. Note also that predicting insignificance (as does Shapiro [28]) is more powerful than predicting significance (as do Lewis and Knowles [27]) since the latter does not lead in general to an exponentially growing tree of coefficients (e.g., edges are more or less one pixel wide in all subbands).

A vectorial algorithm that is close to the iterative refinement zerotree approach of Shapiro used a multistage lattice to progressively refine the vectors of coefficients [125]. Each input vector was coded with a series of vectors of decreasing magnitudes $\{||y_j|| = a_j||x||; j = 1, 2, \cdots\}$, where the following are defined:

j number of coding stages analogous to the number of passes through the dominant and subordinate lists in Shapiro's algorithm

$||y_j||$ magnitude of the reconstruction codevector at stage j

$a < 1$ approximation scaling factor

$||x||$ magnitude of the input vector.

At each stage, the orientation of the reconstruction vector was selected from a finite set of unit energy codevectors. This so-called orientation codebook was designed based on a regular lattice (D_4, E_8, Λ_{16}). Higher dimensional codebooks resulted in slightly better PSNR values. At 0.4 b/pixel, the algorithm outperformed JPEG by 2.5 dB and outperformed the EZW algorithm by a small amount. Further analysis and experimental results are presented in [126]. The algorithm was also applied to the interframe prediction error signal resulting from an overlapped block matching motion compensation technique [127].

Another scheme that uses interband information for intraband coding in the spirit of Shapiro's algorithm but extended to video is presented in [128]. The quantization is scalar and embedded, and zerotree structure and conditional entropy coding take interband information into account.

Kossentini *et al.* incorporated interband information into their multistage RVQ subband system of [79] by using high-order conditional entropy coding conditioned within subbands, across subbands, and across VQ stages [129]. By jointly optimizing the overall quantizer structure within and across subbands while constraining both entropy and complexity, explicit bit allocation was avoided. As in their earlier work, the Lagrangian formulation was used to incorporate the entropy constraint. The product codebooks of the RVQ were searched using tree-structured searching techniques and not in the usual greedy fashion of a multistage encoder. The complexity was controlled by using small alphabets in the individual stage quantizers. Although described for general VQ, the reported results focus on scalar residual quantizers used with a three-level balanced tree-structured IIR all-pass polyphase filter bank with 64 uniform subbands. PSNR's were reported for Lena of 37 dB at 0.5 b/pixel and 34.1 dB at 0.25 b/pixel. This approach was extended to a video codec [130] by incorporating motion-compensation prediction and a form of universal coding. Two coders simultaneously encode the input: one using motion-compensation prediction within the subbands and the other using purely intraframe subband coding. The coder with the minimum Lagrangian is selected and specified by side information to the decoder. An average PSNR of 35.13 dB was reported for the *Miss America* test video sequence at 15.98 kb/s.

In [131], PTSVQ's were designed for each of the 13 bands produced by a four-stage octave-band decomposition with Daubechies eight-tap filters. The baseband was scalar quantized, and the next three bands were quantized as 2-D vectors. Before encoding these three bands, one bit of side information for each vector was sent to the decoder to tell it whether the vector and its descendants should be treated as a zerotree. If so, those vectors were never encoded. Otherwise, the 2-D vectors in those small bands were coded, and at the next level of the pyramid, additional bits of side information were sent to the decoder to say whether the 4-D vectors at the next level and their descendants should be treated as zerotrees or not, and so forth. The higher levels of decomposition used 8-D and 16-D vectors. In one version of the algorithm, the zerotree decision was made by hard thresholding, and in another version, it was decided by a distortion/rate tradeoff. A PSNR of 30.15 was obtained for Lena at 0.174 b/pixel.

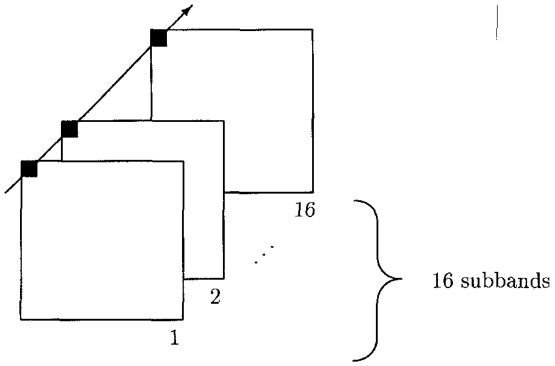


Fig. 7. Crossband vector from uniform subbands.

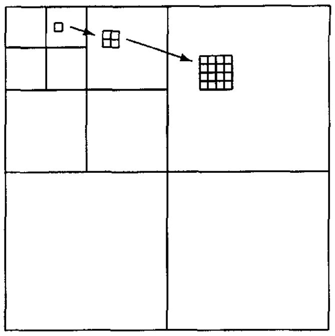


Fig. 8. Same orientation crossband vector from octave-band subbands.

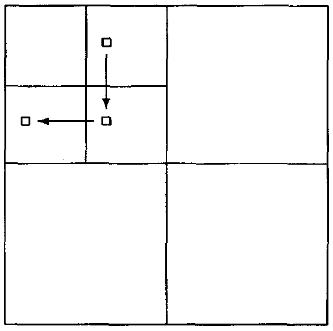


Fig. 9. Same level crossband vector from octave-band subbands.

C. Crossband Vector Methods

Vectors that are put in to the VQ can be formed of coefficients from different subbands. We refer to these as crossband vectors. In a uniform subband decomposition, the crossband vector typically consists of one coefficient from each subband, as shown in Fig. 7; in the example shown, the vector dimension is 16. Sometimes the lowest band will be coded separately, and the crossband vector formed from the other bands. Sometimes, the highest bands will also be excluded from the vector and not coded at all.

In an octave-band decomposition, a crossband vector can be formed out of X_i and its descendants, as shown in Fig. 8. We refer to this as a "same orientation" crossband vector. In the example shown, the vector dimension is 21. Clearly, the dimension of the "same orientation" crossband vector can become large very fast. The octave-band decomposition can also lead to a "same level" crossband vector, as shown in Fig. 9; here, the dimension is 3. For both the "same orientation" and "same level" crossband vectors, there is no natural vector to include the coefficients from the low band. These coefficients could be tacked on individually to any one of the crossband vectors, which would increase the dimension by one. However, recognizing that the low band has very different characteristics, both perceptually and statistically, from the other bands, most subband/VQ methods that employ a crossband vector-forming strategy choose to code the low band separately.

With all three types of crossband vectors, the idea of a weighted distortion measure which accords different weights to different components is appealing since the coefficients of the vector are not all of the same type. This is true both for the "same orientation" vectors, whose components come from different levels of the hierarchy, and for the other two types for which the components come from the same level. In either case, the components have different statistics and different perceptual importance. For instance, the diagonal orientation is considered to be of less perceptual importance than the horizontal or vertical.

1) Crossband Vectors from Uniform Subbands: The first study of crossband VQ's was by Westerink *et al.* [132]. After splitting the image into 16 uniform subbands by 32-tap separable QMF's, 16-D vectors were formed as in Fig. 7. Full-search unstructured VQ's were designed using the GLA. The paper also analyzed the coding gain of VQ over SQ and briefly examined the effect of training sequence size on VQ performance.

Entropy-constrained VQ (ECVQ) and entropy-constrained predictive VQ (ECPVQ) were examined for crossband vectors in [133]. The uniform 16-band decomposition from separable QMF's was encoded with a variety of vector-forming strategies, including SQ separately for all bands, 16-D crossband vectors, and 15-D crossband vectors with the low band coded separately. Different types of entropy coding that were tailored to various possible buffering constraints were considered. Of the systems examined, the most promising used SQ's separately for the 15 high-frequency bands and used ECPVQ on 2-D vectors for the low band. This yielded a PSNR of 30 dB at 0.3 b/pixel for the Lena test image.

In [134], separable 32-tap QMF's were used to split the image into 16 subbands. The coefficients were encoded as 16-D crossband vectors by finite-state VQ. The *master* codebook had 256 codewords determined by the GLA. For each input vector x, let u and l denote the input vectors whose coefficients lie directly above and directly to the left, respectively, of the coefficients that compose x. For each x, the *state* codebook is composed of the 16 codewords m from the master codebook that have the smallest value of $distortion(m, u) + distortion(m, l)$, where the distortion is MSE. Since the decoder has already decoded u and l before it receives x, it is able to determine the same state codebook as the encoder. The input vector x is encoded by finding the nearest neighbor in this state codebook. If the distortion produced by this nearest neighbor is greater than some threshold, then the nearest neighbor from the master codebook is chosen instead, and side information must be sent to the decoder to inform it of this change. This limits the maximum distortion and can prevent the finite-state machine from becoming derailed. The method provided PSNR's of 30–34 dB for images Peppers, Airplane, and Zelda at 0.3 b/pixel.

2) "Same-Orientation" Crossband Vectors from Octave-Band Subbands: Huh *et al.* used classified VQ on the coefficients resulting from a three-stage wavelet decomposition using Johnston's near-PR filters [135]. For purposes of classification, the 21-D vectors as shown in Fig. 8 were further grouped together. Those vectors corresponding to the same spatial location but different orientations in the original image, including the single coefficient from the lowest band, formed a 64-D group (21 + 21 + 21 + 1 = 64). These 8 × 8 blocks were categorized into four classes, depending on the directional activity: horizontally active, vertically active, diagonally active, inactive. This class information was sent separately to the decoder. Depending on the class, certain of the 64 coefficients were discarded altogether, with the remaining ones divided into between three and seven smaller subvectors of dimensions ranging from 1 to 11. Only one subvector of one class actually retained coefficients that cross bands. Therefore, this method is essentially one that used the interband correlation for purposes of classification but did not code the coefficients as crossband vectors. The algorithm produced PSNR's of 34.1, 32.3, and 30.2 for Lena at 0.5, 0.33, and 0.21 b/pixel, respectively.

The same research group tried this classification method with two-channel conjugate VQ [136], [137]. Conjugate VQ is a structure that can save on both memory and computational complexity. Where a regular VQ might employ a single codebook with N^2 vectors, a two-channel conjugate VQ allows N^2 possible reproduction vectors using two codebooks (A and B) with N vectors each. Each reproduction vector is made from some combination of a codeword from A and a codeword from B. As before, coefficients from the 10 bands were grouped into 8 × 8 blocks for classification into the four classes. Based on the class, the blocks were divided into smaller subvectors for encoding with two-channel conjugate VQ. A PSNR of 31.8 dB was obtained at 0.32 b/pixel for the Lena test image.

Vectors of dimension 64 were also examined in [138], where the 21-D vectors as shown in Fig. 8 were grouped together from different orientations, including the single coefficient from the lowest band. The method used nonlinear interpolative VQ [139] in which a high-dimensional vector is encoded using a size-reduced feature vector, and the decoder reconstructs the high-dimensional vector using an optimal nonlinear interpolation. In [138], the reduced-size feature vector was composed of the 16 low-frequency elements out of the 64. A biorthogonal filter with three to five taps was used. The 16-D encoder codebook was generated by using the GLA on a training sequence, and the 64-D decoder codebook was designed by projecting the resultant subspaces of the 16-D vector space into the 64-D vector space. The decoder and encoder codebooks would then have the same number of codewords. A PSNR of 31.78 dB was obtained at 0.203 b/pixel for the 256 × 256 Lena test image.

A variable-dimension same-orientation VQ scheme was proposed in [140] for use in intraframe coding situations of high motion and scene changes. The decomposition was into 10 bands using a biorthogonal five- to seven-tap wavelet basis. Starting with a 21-D same-orientation vector as shown in Fig. 8, the coefficients were "merged" in a bottom-up quadtree fashion as described in [141], in which the decision to merge depended on whether any of the differences between neighboring coefficients exceeded a predetermined threshold. The resulting vector could have dimension 5, 9, 13, 17, or 21, and was encoded with a weighted distortion measure using a codebook of the appropriate dimension which had been designed by the GLA.

In [142], a three-level octave-band decomposition from rectangular separable Gabor-like wavelets was used with 21-D same-orientation crossband vectors. Those vectors with all components below a specified threshold were designated zerotrees and not coded. The remaining coefficients were reorganized into 5-D interband vectors (one coefficient from the coarser level and four from the next) and 16-D intraband vectors (4 × 4 blocks from the finest bands). The 5-D interband vectors were encoded using a weighted distortion measure, and the intraband vectors were encoded using an unweighted full-search VQ. The method was extended to color images by separately coding luminance and chrominance coefficients, discarding lower subbands of the chrominance components, using a weighted quadratic distortion measure based on perceptual sensitivity, and using a classified VQ [143]. Two classification strategies were adopted: one based on subbands with common frequency orientation and the other on separating edge, texture, and homogeneous regions. A PSNR on the luminance component of 28.19 dB was achieved at 0.14 b/pixel, but it should be kept in mind that code optimization was with respect to a weighted distortion and not the MSE inherent in SNR.

3) "Same-Level" Crossband Vectors from Octave-Band Subbands: Vectors were formed as in Fig. 9 in [144]. A 24-tap QMF was used to decompose images into seven subbands. The baseband was coded using predicted mean/residual VQ on 2×2 blocks. The previously encoded blocks were used to make a prediction of the mean value of the next block to be encoded. This predicted mean value was removed from the block to be encoded. The decoder was able to make the same prediction using its available encoded blocks. The prediction was added back to the decoded vector to form the reconstruction. A truncated Laplacian of Gaussian operator $\nabla^2 G$ was applied by both the encoder and the decoder to the reconstructed baseband image. The zero crossings of this filtered subimage indicated the positions of the edges. Only those coefficients from the medium and high bands that correspond to edge locations were coded. In one scheme, a 3-D vector was formed from the three medium bands by taking the corresponding pixels from the three different orientations. In another scheme, the residual baseband subimage was formed by taking the difference between the actual and the reconstructed baseband. Then, a 4-D vector was formed by adding the corresponding coefficient from this residual image to the 3-D vector of the previous scheme. In either case, the medium bands were dequantized and inverse transformed to obtain a baseband at the next level. The edge detector was applied to the new baseband in order to determine the edge positions to be encoded in the high bands. The same method was applied to image sequences, where block matching was used to estimate block motion, and the subband filters were then applied to the motion-compensated frame difference image [144], [145]. Results for Lena were a PSNR of 34.1 at a bit rate between 0.69 and 0.50 b/pixel, depending on whether the bit rate was calculated as the fixed rate of a constant wordlength code for the codebook addresses or as the entropy of a variable wordlength code.

In a series of two articles, Buhmann and Kühnel developed the notion of VQ with complexity costs, which they referred to as entropy optimized VQ, and demonstrated its use for coding wavelet coefficients [146], [147]. The basic idea is a variation on ECVQ, but the formulation is more general and the specific algorithm different. The authors noted that the traditional VQ design strategy, as followed by the GLA, relies exclusively on the distortion errors for placing the reproduction vectors. The number of reproduction vectors is assumed to be chosen in advance. By incorporating into the cost function an application-dependent complexity measure, the optimization of distortion and the intrinsic constraints on the codebook can be allowed to affect each other in some appropriate fashion. The test images were decomposed with a separable QMF filter to yield seven subbands. The lowest band was scalar quantized. The remaining bands were formed into 3-D vectors at each level, as shown in Fig. 9. GLA clustering was compared against entropy-optimized VQ, with the latter resulting in significantly higher PSNR's on test images. The GLA method produced considerably smaller codebooks than the entropy optimized VQ. The latter was able to place reproduction vectors in sparsely populated areas, preserving psychovisually important image features like edges more faithfully. In the experiments, the two methods were required to have the same output *entropy* rather than the same output *bit rate* (in which case, ECVQ may have been the more appropriate benchmark).

A type of "same-level" crossband vector was used in [148], where the image was decomposed into even and odd symmetric size and orientation-selective bandpass filter outputs. At each level of resolution, an 8-D vector space resulted from a polar representation of local amplitude and local phase from four orientation-selective analytic bandpass filters. This hyperspace was partitioned in a feature-specific way, taking into account properties of the human visual system. The small hypercube at the origin was assumed to correspond to an irrelevance zone conforming to the threshold properties of human vision. All vectors in that region could be quantized to the origin. The regions around the hypersphere coordinate axes represented intrinsically 1-D image features, as only one orientation filter was giving a strong response. Regions between two axes were differentiated into those corresponding to 1-D image features whose orientations did not exactly coincide with the preferred axis of an orientation filter and those that corresponded to certain intrinsically 2-D features such as discontinuities. These regions needed to be quantized relatively finely, and a saturation function in accordance with masking effects was applied. Vectors in the residual space arising from intrinsically 2-D image features such as curved lines and textural features that activate more than two orientation filters could be quantized more coarsely. The quantization step thus involved only simple decisions and nonlinear transformations followed by uniform quantization with no codebook search or distance calculations. The VQ yielded an uneven probability density function for the vectors, and an ideal entropy coder was assumed. Satisfactory subjective image quality was found at 0.78 b/pixel for the 256 × 256 Lena test image.

A crossband gain/shape VQ used for a 64-band decomposition was briefly described in [34], although this paper is primarily concerned with elaborating upon an error-resilient technique for efficiently encoding positional information. In two papers, SQ was used on wavelet coefficients, and crossband vectors were used to design subsequent entropy coders [149], [150]. In [149], the output of a six-band quasiperfect reconstruction filter was quantized with SQ followed by arithmetic coding, and a crossband VQ was used to provide the arithmetic coder with a probability density function. In [150], SQ was applied to the coefficients of a 10-band pyramidal decomposition where the FIR filter banks were allowed to be different at each level of the decomposition. The images decomposed were motion-compensated progressive frames derived from a novel adaptive block matching algorithm. The interband information was used to design a universal variable length entropy coder for vectors that were composed of the SQ output values.

4) Wavelet Hierarchical VQ (WHVQ): Vishwanath and Chou [70] realized that the subband decomposition and VQ could be accomplished together by table lookups using the HVQ structure. This provides a number of benefits: natural scalability, ease of incorporation of perceptual considerations into the code design, and a very fast software implementation that requires only table lookups and no computation for the decomposition, quantization, and reconstruction. There is significant computation off line when the tables are filled since, as with ordinary HVQ, the tables simply capture the result of a complicated mapping from one discrete space into another.

The system uses separable subband filters and operates alternately on rows and columns. Each stage in the hierarchical coder does one row or one column decomposition along with a VQ. In this way, quantization is done across the subbands of the wavelet transform. The system is best suited to short filters to concentrate their effects within the vector size used by the VQ. As with ordinary HVQ, each stage will combine pairs of indexes to form individual indexes, providing a 2:1 compression. Unlike ordinary HVQ, these tables do not simply implement the result of nearest neighbor searches. They also include filtering and downsampling prior to the quantization. Each stage consists of two steps. The first is simply an HVQ on the preceding stage, which will be raw pixels in the first stage and quantized wavelet coefficients thereafter. The first step, providing 2:1 compression, successively maps nonoverlapping pairs of k-dimensional blocks produced by the previous stage into a single quantized block corresponding to a $2k$-dimensional vector. The second step maps two of the first-step output blocks into a stage-2 output block but does so using overlapping blocks because the wavelet decomposition requires the larger window. This step is designed by clustering the reproduction vectors of the previous stage in a sliding-block manner, each time shifting the window by k.

Following each stage, downsampling and necessary delays are easily done without computation. Each stage operates in only one dimension so that rows and columns are handled by alternating stages. The overall effect of a single stage is to produce wavelet coefficients and to compress by 2:1.

The system is inherently fixed rate, and no entropy constrained versions exist for optimizing for entropy coding. Using the Daubechies-4 filter and the 256 × 256 Lena image, the reported PSNR was 29.62 dB for 0.5 b/pixel and 27.26 dB for 0.25 b/pixel. On a Sparc 2 with 16M of memory, encoding and decoding took 16.67 ms for the 0.5 b/pixel image and 19 ms for the 0.25 b/pixel image. The encoding required 64 KB of memory per stage.

D. Subband VQ Methods for Color Images

The great potential for efficient exploitation of properties of the human visual system for monochrome subband/VQ systems is even greater for color subband/VQ systems. The earliest color subband/VQ system was by Kim *et al.* [151], and it did not explicitly incorporate models of human vision, although some such ideas were employed, for example, for deciding which frequency bands to drop. Later subband/VQ systems for color images began to explicitly exploit models of human vision [152], [153] and color vision [154], [155], [105].

The first color subband/VQ system used DPCM together with either CVQ or FSVQ [151]. RGB images were transformed to the YIQ color system, and each color plane was filtered by symmetrically extended QMF's to produce 16 uniform frequency bands. For Q and I planes, only bands Q_{11}, I_{11}, I_{12} and I_{21} were retained. The 12 lowest frequency bands of the luminance plane were retained, producing a total of 16 bands to be encoded. The low bands (Y_{11}, I_{11}, Q_{11}) were encoded by DPCM. Different approaches were tried for the higher bands. Each 4 × 4 vector was coarsely classified as being shade, high variance, or one of several edges. This classification was sent to the decoder as side information, and the appropriate codebook was used. An improved version used an FSVQ to select the codebook in a manner very similar to that of Aravind and Gersho [63]. In a continuation of this work, Kim *et al.* used a three-component multirate representation of the image composed of a chrominance subimage pair, a medium- to low-frequency luminance image, and a high-frequency luminance residual. The method used FSVQ and DPCM in coding these components with noticeable improvement over the earlier work [156].

Safranek *et al.* presented two examples of perceptually optimized coding algorithms [152]. Variable-rate DPCM coding together with a model in which the quantization threshold for just noticeable distortion is a function of three parameters (center frequency or subband number, background intensity, and background texture) led to transparent coding of color images at rates ranging from 0.2 to 2.0 b/pixel, depending on the input. For high-quality coding of sparse images, such as high-frequency subbands in 3-D coding of video signals, a geometrical VQ was used whose codevectors were edge-related shapes rather than traditional designs based on minimum MSE. In the coding of teleconference scenes, average bit rates for high-frequency subbands were in the range of 0.01 to 0.3 b/pixel.

Spatial frequency dependence of the human visual system was incorporated into the VQ distortion measure in [153]. This work used two complementary thresholds, known as the visibility threshold and the annoyance threshold, in the global evaluation of the distortion during the splitting design of the codebook for each class. These techniques claimed to yield visual VQ of the chrominance component of a color video digital signal with no visible impairment and with a high compression ratio.

Full search of an unstructured codebook was examined for color images in several studies. In [157], 2-D uniform band decomposition was applied to the error image obtained after motion-compensated interframe prediction. The luminance error image was decomposed into 16 subbands and the chrominance error image into eight subbands. These were limited to 16 bands total by taking 10 luminance bands and six chrominance bands, discarding the lower energy ones. Sixteen-dimensional crossband vectors were formed as in Fig. 7 and quantized by full search of an unstructured codebook designed by the GLA. Each subband location has an "area of influence" of size 5 × 5 in the original image. The subband location was considered significant if the percentage of nonzero pixels in its area of influence was greater than a predetermined threshold. The insignificant subband locations were run-length coded, and the subband coefficients from the significant locations were vector quantized. This sample selection scheme can help prevent buffer overflow.

In another study using full-search unstructured VQ, Van Dyck and Rajala [154], [155] formulated the bit allocation problem as an optimization problem where the objective function depended on the distortion-rate curves of the quantizers and on a set of perceptual weights. These weights were derived from data provided by measurements of the mean detection threshold of the human visual system for color transitions along the luminance, red-green, and blue-yellow directions. Minimization of the objective function constrained by the desired bit rate gave a perceptually optimal bit allocation. Three subband/VQ cases were examined. In all cases, the color image was decomposed two levels by a QMF (32-tap filter D) to form seven subbands. In the first two cases, the color components of the lowest frequency subband were scalar quantized. Case 1 combined the three color components of each pixel of the higher frequency subbands into a 3-D vector, whereas case 2 created 4-D vectors from 2 × 2 blocks in each subband color component. In the third case, the chrominance components of the lowest frequency subband were also vector quantized with 2 × 2 blocks. In all cases, a full-search VQ was generated by the GLA. To obtain the required compression ratio and the high color fidelity required for HDTV applications, the VQ was done in the perceptually uniform C.I.E. L*a*b* [158] space and AC_1C_2 space. Subsequently, Van Dyck and Rajala used VQ with a separable diamond subband coder in order to take into account the higher sensitivity of the human eye to horizontal and vertical edges than to diagonal ones [105]. By first diagonally interpolating the input image and then filtering with 1-D filters, a decomposition into subbands that more closely match the orientation of the human visual system was obtained. Performance evaluations showed that for compression ratios in the range of 10:1 to 20:1, a five-band system with diamond-shaped subbands yielded results comparable with those of a seven-band rectangular system.

A comparison of ECVQ and ECTCQ for color video sequences was conducted in [159]. The RGB format was converted to YUV. The luminance component underwent a seven-band decomposition, and each chrominance component was decomposed into four bands. The lowest frequency subband used a variant of the H.261 coder, and the higher frequency bands were coded either with ECVQ on 2×2 blocks from a particular subband color component or with ECTCQ. The authors concluded that the system could provide good quality video for moderately complex scenes at around 400 kb/s at 10 frames/s.

In [160], a "same-orientation" crossband vector was formed after three levels of decomposition with a Gabor-like wavelet filter applied to a color image in the YUV color space. For each color plane, the lowest band was scalar quantized. For the next higher bands, 15-D vectors were formed by taking five coefficients from each color plane: One coefficient from the lower band combined with the 2×2 block from the next higher band of the same orientation. At the highest level, the UV color planes were discarded, and the luminance plane was encoded as 4×4 intraband blocks. Unstructured full-search codebooks were designed by the GLA for each group of vectors. A PSNR of 28.2 dB was achieved at 1.13 b/pixel for the color Lena test image.

V. Conclusions

Subband decompositions and VQ are two powerful tools for image compression. Recent years have seen an explosion in the research efforts to put the two tools together in useful ways. VQ applied to subband coefficients raises some issues that are distinctive and not shared with VQ applied to raw pixel data or other types of transform coefficients. Some of these issues include the following:

- In the image domain, strategies for forming vectors are relatively simple. Image pixels that are closer together are more highly correlated, and a VQ performs better if its input vectors have components that are more highly correlated. Therefore, vectors in the image domain are formed as compact little collections of adjacent pixels. Gains in perceptual quality or efficiency can be attained by tweaking the shape or size of the block [161], but the basic strategy is straightforward. With subband coefficients, correlations exist both intraband and interband, and the best vector-forming strategy is not obvious.
- Similarly, predictive and finite-state methods are more straightforward in the image domain since (for still images) only one type of correlation can typically be exploited. The adjacent blocks that have already been encoded can be used to help encode the next block either with prediction or a finite-state method. In a subband decomposition, predictive and finite-state methods can use either intra or interband information.
- Methods for exploiting properties of human vision are considerably less developed in the image domain than in the frequency domain. While some visual system characteristics, such as textural masking, can and have been exploited by purely image domain VQ, much more sophisticated attempts have been made for coefficients resulting from a DCT or subband decomposition since those representation lend themselves more naturally to the realization of visually based coding. That is, it is rather difficult in the image domain to conclude that some pixels are *a priori* less important than others but with transform coefficients that determination is more easily made.
- Spatial information is diffused with the subband filtering; therefore, coding methods that are based on segmentation or object recognition may be more difficult to implement with a subband transform. However, the hierarchical nature of the decomposition does mean that any sort of segmentation or motion-compensation information that can be extracted from the low bands can perhaps be used "for free" by the decoder since it is usually assumed that the low bands will be transmitted first, and they can be examined by the decoder to aid in the subsequent decoding. For example, lower resolution small blocks can be used to predict the locations of high-resolution larger blocks in a "coarse-fine" searching strategy for block-based motion compensation.
- The multiresolution nature of subband decompositions and the progressive nature of some quantization methods both provide means of making reconstruction quality scalable or progressive. Scalability implies that there is a "successive approximation" property in the bit stream. As the decoder gets more bits from the encoder, the decoder can decode a progressively better reconstruction of the image. This can be achieved by adding subbands (also called layers) to the currently decoded image or by improving coefficients in the subbands. While this added feature is attractive for a number of applications (the most obvious being progressive transmission of images over slow telephone lines), it was long thought that it would lead to less efficient compression. As Shapiro's embedded zerotree algorithm [28] demonstrates, however, scalability can be achieved while being competitive with other approaches. With scalable coding, a single encoder can provide a variety of rates to customers with different channels or display capabilities. Since images can be reconstructed to increasing quality as additional bits arrive, it provides a natural means of adjusting to changing channel capacities and a more effective means of using a relatively slow channel (as in progressive transmission). Improved performance in

scalable coding may be achievable by better melding the decomposition and quantization operations than is currently done. In addition, current scalable coders tend to be computationally more complex than their nonscalable counterparts.

At the current state of the art, subband/VQ coders are very competitive with subband/SQ coders that incorporate sophisticated lossless coding. In some cases, the VQ provides improved distortion/rate tradeoffs at the expense of increased complexity. Performance improvements continue to be reported on both sides. The ability of VQ to incorporate other signal processing into the compression process has not yet been well exploited. Although Shannon theory supports the distortion/rate superiority of VQ over SQ, much current work is focusing on exploiting properties of human vision and dependencies across scales, and it remains to be seen whether VQ algorithms can achieve the advantages predicted by theory while exploiting these properties in a flexible and low-complexity way. It is not unlikely that future generations of image coding standards will incorporate subband or wavelet decompositions and SQ or VQ. The future gains in subband coding are perhaps more likely to result from improved quantizing of the coefficients rather than from improvements in the filtering techniques themselves.

Acknowledgment

The authors gratefully acknowledge helpful comments by S. Bedros, P. Chou, M. Cohn, and J. Villasenor.

References

[1] P. C. Cosman, K. L. Oehler, E. A. Riskin, and R. M. Gray, "Using vector quantization for image processing," *Proc. IEEE*, vol. 81, no. 9, pp. 1326–1341, Sept. 1993.

[2] Y. Qi and B. R. Hunt, "A multiresolution approach to computer verification of handwritten signatures," *IEEE Trans. Image Processing*, vol. 4, no. 6, pp. 870–874, June 1995.

[3] J. S. Baras and S. I. Wolk, "Model based automatic target recognition from high range resolution radar returns," in *Proc. Automat. Object Recogn. IV*, vol. 2234, 1994, pp. 57–66.

[4] J. W. Woods, Ed., *Subband Image Coding*. Boston: Kluwer, 1991.

[5] M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[6] O. Rioul and M. Vetterli, "Wavelets and signal processing," *IEEE Signal Processing Mag.*, vol. 8, no. 4, pp. 14–38, Oct. 1991.

[7] G. Strang, "Wavelet transforms versus fourier transforms," *Bull. (New Series) AMS*, vol. 28, no. 2, pp. 288–305, Apr. 1993.

[8] I. Daubechies, *Ten Lectures on Wavelets*. Philadelphia: Soc. Ind. Appl. Math., 1992.

[9] G. Kaiser, *A Friendly Guide to Wavelets*. Boston: Birkhäuser, 1994.

[10] A. Croisier, D. Esteban, and C. Galand, "Perfect channel splitting by use of interpolation/decimation/tree decomposition techniques," in *Int. Conf. Inform. Sci. Syst.*, Patras, Greece, Aug. 1976, pp. 443–446.

[11] R. E. Crochiere, S. M. Webber, and J. K. L. Flanagan, "Digital coding of speech in subbands," *Bell Syst. Tech. J.*, vol. 55, pp. 1069–1086, Oct. 1976.

[12] M. Vetterli, "Multi-dimensional sub-band coding: Some theory and algorithms," *Signal Processing*, vol. 6, pp. 97–112, Apr. 1984.

[13] J. W. Woods and S. D. O'Neil, "Subband coding of images," *IEEE Trans. Acoust. Speech Signal Processing*, vol. ASSP-34, pp. 1278–1288, Oct. 1986.

[14] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Commun.*, vol. COM-31, pp. 532–540, Apr. 1983.

[15] R. R. Coifman and M. V. Wickerhauser, "Entropy-based algorithms for best basis selection," *IEEE Trans. Inform. Theory*, vol. 38, no. 2, pp. 713–718, 1992.

[16] J. D. Johnston, "A filter family designed for use in quadrature mirror filter banks," in *Proc. ICASSP*, Denver, CO, 1980, pp. 291–294.

[17] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Commun. Pure Appl. Math.*, vol. 41, pp. 909–996, Nov. 1988.

[18] A. Cohen, I. Daubechies, and J.-C. Feauveau, "Biorthogonal bases of compactly supported wavelets," *Commun. Pure Applied Math.*, vol. 45, pp. 485–560, 1992.

[19] M. Vetterli and C. Herley, "Wavelets and filter banks: Theory and design," *IEEE Trans. Signal Processing*, vol. 40, no. 9, pp. 2207–2232, Sept. 1992.

[20] J. D. Villasenor, B. Belzer, and J. Liao, "Wavelet filter evaluation for image compression," to appear.

[21] P. Onno and C. Guillemot, "Tradeoffs in the design of wavelet filters for image compression," in *Proc. Visual Commun. Image Processing '93*, vol. 2094, 1993, pp. 1536–1547.

[22] J. P. Andrew, P. O. Ogunbona, and F. J. Paoloni, "Coding gain and spatial localization properties of discrete wavelet transform filter banks for image coding," in *Proc. ICIP-94*, Austin, TX, vol. 3, Nov. 1994, pp. 348–352.

[23] M. G. Strintzis, "Optimal selection of multi-dimensional biorthogonal wavelet bases," in *Proc. ICIP-94*, Austin, TX, vol. 3, Nov. 1994, pp. 353–357.

[24] J. Katto and Y. Yasuda, "Performance evaluation of subband coding and optimization of its filter coefficients," in *Proc. Visual Commun. Image Processing '91*, vol. 1605, 1991, pp. 95–106.

[25] R. A. DeVore, B. Jawerth, and B. Lucier, "Data compression using wavelets: error, smoothness, and quantization," in *Proc. IEEE Data Compression Conf.*, J. A. Storer and J. H. Reif, Eds., Apr. 1991, pp. 186–195.

[26] R. A. DeVore, B. Jawerth, and B. Lucier, "Image compression through wavelet transform coding," *IEEE Trans. Inform. Theory*, vol. 38, no. 2, pp. 719–746, 1992.

[27] A. S. Lewis and G. Knowles, "Image compression using the 2-D wavelet transform," *IEEE Trans. Image Processing*, vol. 1, no. 2, pp. 244–250, Apr. 1992.

[28] J. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3445–3462, Dec. 1993.

[29] Z. Xiong, K. Ramchandran, and M. T. Orchard, "Joint optimization of scalar and tree-structured quantization of wavelet image decompositions," A. Singh, Ed., in *Conf. Rec. Twenty-Seventh Asilomar Conf. Signals, Syst. Comput.*, 1993, vol. 2, pp. 891–895.

[30] S. P. Lloyd, "Least squares quantization in PCM," Unpublished Bell Labs. Techn. Note; portions presented at the Inst. Math. Stat. Mtg. Atlantic City, NJ, Sept. 1957; published in the Mar. 1982 Special Issue on Quantization of the *IEEE Trans. Inform. Theory*.

[31] J. Max, "Quantizing for minimum distortion," *IEEE Trans. Inform. Theory*, pp. 7–12, Mar. 1960.

[32] O. Johnsen, O. V. Shentov, and S. K. Mitra, "A technique for the efficient coding of the upper bands in subband coding of images," in *Proc. ICASSP*, Albuquerque, NM, USA, Apr. 1990, pp. 2097–2100.

[33] H. Gharavi and A. Tabatabai, "Subband coding of monochrome and color image," *IEEE Trans. Circuits Syst.*, vol. 35, no. 2, pp. 207–214, Feb. 1988.

[34] N.-T. Cheng and N. G. Kingsbury, "The ERPC: An efficient error-resilient technique for encoding positional information or sparse data," *IEEE Trans. Commun.*, vol. 40, no. 1, pp. 140–148, Jan. 1992.

[35] D. L. Donoho, "Unconditional bases are optimal bases for data compression and for statistical estimation," *Appl. Computation. Harmonic Anal.*, vol. 1, no. 1, pp. 100–115, Dec. 1993.

[36] R. J. Safranek and J. D. Johnston, "A perceptually tuned sub-band image coder with image dependent quantization and post-quantization data compression," in *Proc. ICASSP*, Glasgow, UK, 1989, pp. 1945–1948.

[37] L. Vandendorpe and B. Macq, "Optimum quantization for subband coders," in *Proc. Visual Commun. Image Processing '90*, Lausanne, Switzerland, vol. 1360, 1990, pp. 898–908.

[38] P. H. Westerink, J. Biemond, and D. E. Boekee, "An optimal bit allocation algorithm for sub-band coding," in *Proc. ICASSP*, 1988, pp. 757–760.

[39] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Trans. Acoust. Speech Signal Processing*, vol. 36, no. 9, pp. 1445–1453, Sept. 1988.

[40] 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, no. 2, pp. 299–315, Mar. 1989.

[41] E. A. Riskin and R. M. Gray, "A greedy tree growing algorithm for the design of variable rate vector quantizers," *IEEE Trans. Signal Processing*, vol. 39, pp. 2500–2507, Nov. 1991.

[42] E. A. Riskin, "Optimal bit allocation via the generalized BFOS algorithm," *IEEE Trans. Inform. Theory*, vol. 37, pp. 400–402, Mar. 1991.

[43] T. Senoo and B. Girod, "Vector quantization for entropy coding of image subbands," *IEEE Trans. Image Processing*, vol. 1, no. 4, pp. 526–532, Oct. 1992.

[44] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Boston: Kluwer, 1992.

[45] J.-Y. Huang and P. M. Schultheiss, "Block quantization of correlated Gaussian random variables," *IEEE Trans. Commun.*, vol. COM-11, pp. 289–296, Sep. 1963.

[46] H. Gish and J. N. Pierce, "Asymptotically efficient quantizing," *IEEE Trans. Inform. Theory*, vol. IT-14, pp. 676–683, Sept. 1968.

[47] A. Gersho, "Asymptotically optimal block quantization," *IEEE Trans. Inform. Theory*, vol. IT-25, pp. 373–380, July 1979.

[48] 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.

[49] H. Abut, Ed., *Vector Quantization*, IEEE Reprint Collection. Piscataway, NJ: IEEE, May 1990.

[50] N. M. Nasrabadi and R. A. King, "Image coding using vector quantization: A review," *IEEE Trans. Commun.*, vol. 36, pp. 957–971, Aug. 1988.

[51] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Entropy-constrained vector quantization," *IEEE Trans. Acoust. Speech Signal Processing*, vol. 37, pp. 31–42, Jan. 1989.

[52] R. M. Gray, *Source Coding Theory*. Boston: Kluwer, 1990.

[53] J. H. Conway and N. J. A. Sloane, "Fast quantizing and decoding algorithms for lattice quantizers and codes," *IEEE Trans. Inform. Theory*, vol. 28, pp. 227–232, Mar. 1982.

[54] ——, *Sphere Packings, Lattices and Groups*. New York: Springer-Verlag, 1988.

[55] M. J. Sabin and R. M. Gray, "Product code vector quantizers for waveform and voice coding," *IEEE Trans. Acoust. Speech Signal Processing*, vol. ASSP-32, pp. 474–488, June 1984.

[56] B.-H. Juang and A. H. Gray, Jr., "Multiple stage vector quantization for speech coding," in *Proc. ICASSP*, Paris, vol. 1, Apr. 1982, pp. 597–600.

[57] Y.-S. Ho and A. Gersho, "Variable-rate multi-stage vector quantization for image coding," in *Proc. ICASSP*, Apr. 1988, pp. 1156–1159.

[58] W. Y. Chan, S. Gupta, and A. Gersho, "Enhanced multistage vector quantization by joint codebook design," *IEEE Trans. Commun.*, vol. 40, no. 11, pp. 1693–1697, Nov. 1992.

[59] C. F. Barnes and R. L. Frost, "Necessary conditions for the optimality of residual vector quantizers," in *Abs. 1990 IEEE Int. Symp. Inform. Theory*, San Diego, CA, USA, Jan. 1990, p. 34.

[60] R. L. Frost, C. F. Barnes, and F. Xu, "Design and performance of residual quantizers," in *Proc. Data Compression Conference*, J. A. Storer and J. H. Reif, Eds., Snowbird, UT, USA, Apr. 1991, pp. 129–138.

[61] C. F. Barnes and R. L. Frost, "Vector quantizers with direct sum codebooks," *IEEE Trans. Inform. Theory*, vol. 39, no. 2, pp. 565–580, Mar. 1993.

[62] B. Ramamurthi and A. Gersho, "Classified vector quantization of images," *IEEE Trans. Commun.*, vol. COM-34, no. 11, pp. 1105–1115, Nov. 1986.

[63] R. Aravind and A. Gersho, "Low-rate image coding with finite-state vector quantization," in *Proc. ICASSP*, Tokyo, 1986, pp. 137–140.

[64] T. Kim, "New finite state vector quantizers for images," in *Proc. ICASSP*, New York, NY, USA, Apr. 1988, pp. 1180–1183.

[65] G. Ungerboeck, "Trellis-coded modulation with redundant signal sets, Parts I and II," *IEEE Commun. Mag.*, vol. 25, pp. 5–21, Feb. 1987.

[66] T. R. Fischer, M. W. Marcellin, and M. Wang, "Trellis-coded vector quantization," *IEEE Trans. Inform. Theory*, vol. 37, pp. 1551-1566, Nov. 1991.

[67] M. W. Marcellin and T. R. Fischer, "Trellis coded quantization of memoryless and Gauss-Markov sources," *IEEE Trans. Commun.*, vol. 38, pp. 82–93, 1990.

[68] M. W. Marcellin, T. R. Fischer, and J. D. Gibson, "Predictive trellis coded quantization of speech," *IEEE Trans. Acoust. Speech Signal Processing*, vol. 38, pp. 46–55, Jan. 1990.

[69] P. C. Chang, J. May, and R. M. Gray, "Hierarchical vector quantizers with table-lookup encoders," in *Proc. 1985 IEEE Int. Conf. Commun.*, vol. 3, June 1985, pp. 1452–1455.

[70] M. Vishwanath and P. Chou, "An efficient algorithm for hierarchical compression of video," in *Proc. ICIP-94*, Austin, TX, vol. III, Nov. 1994, pp. 275–279.

[71] N. M. Akrout, C. Diab, R. Prost, and R. Goutte, "Codeword orientation: an improved subband coding/vector quantization scheme for image coding," *Opt. Eng.*, vol. 33, no. 7, pp. 2394–2398, July 1994.

[72] D. Barba and A. Neudecker, "Psychovisual lattice vector quantization in subband image coding," in *Proc. Visual Commun. Image Processing '93*, vol. 2094, 1993, pp. 929–940.

[73] Z. Gao, F. Chen, B. Belzer, and J. Villasenor, "A comparison of the Z, E_8, and Leech lattices for image subband quantization," in *Proc. 1995 IEEE Data Compression Conf.*, J. A. Storer and M. Cohn, Eds., Snowbird, UT, USA, Mar. 1995, pp. 312–321.

[74] R. J. Safranek, K. MacKay, N. S. Jayant, and T. Kim, "Image coding based on selective quantization of the reconstruction noise in the dominant sub-band," in *Proc. ICASSP*, New York, NY, vol. 3, Apr. 1988, pp. 765–768.

[75] I. Furukawa, M. Nomura, and S. Ono, "Hierarchical coding of super high definition images with subband + multistage VQ," in *Proc. ICASSP*, Toronto, Canada, May 1991, pp. 2637–2640.

[76] C. F. Barnes and E. J. Holder, "Classified variable rate residual vector quantization applied to image subband coding," in *Proc. 1993 IEEE Data Compression Conf. (DCC)*, J. A. Storer and M. Cohn, Eds., Snowbird, UT, USA, Mar. 1993, pp. 272–281.

[77] I. Furukawa, M. Nomura, N. Ohta, and S. Ono, "Hierarchical coding of super high definition images with adaptive block-size multi-stage VQ," in *Signal Processing HDTV III. Proc. Fourth Int. Workshop HDTV Beyond*, Turin, Italy, Sept. 1991, pp. 361–368.

[78] I. Furukawa, M. Nomura, and S. Ono, "Hierarchical sub-band coding of super high definition image with adaptive block-size multistage VQ," *Signal Processing: Image Commun.*, vol. 5, no. 5–6, pp. 527–538, Dec. 1993.

[79] F. Kossentini, W. C. Chung, and M. J. T. Smith, "Subband image coding using entropy-constrained residual vector quantization," *Inform. Processing Mgmt.*, vol. 30, no. 6, pp. 887–896, 1994.

[80] C. K. Chui and C. Li, "Non-orthogonal wavelet packets," *SIAM J. Math. Anal.*, vol. 24, pp. 712–738, 1993.

[81] Q. Liu, A. K. Chan, C. K. Chui, E. Pettit, and D. Rhines, "A hybrid technique using spline-wavelet packets and vector quantization for high rate image compression," in *Math. Imaging*, vol. 2034, 1993, pp. 194–204.

[82] E. S. Jang and N. M. Nasrabadi, "Subband coding with multi-stage VQ for wireless image communication," *IEEE Trans. Video Tech.*, vol. 5, no. 3, 1995.

[83] M. Antonini, M. Barlaud, and P. Mathieu, "Image coding using lattice vector quantization of wavelet coefficients," in *Proc. ICASSP*, Toronto, Canada, May 1991.

[84] M. Barlaud, P. Solé, M. Antonini, and P. Mathieu, "A pyramidal scheme for lattice vector quantization of wavelet transform coefficients applied to image coding," in *Proc. ICASSP*, vol. 4, Mar. 1992, pp. 401–404.

[85] M. Barlaud, P. Solé, T. Gaidon, M. Antonini, and P. Mathieu, "Pyramidal lattice vector quantization for multiscale image coding," *IEEE Trans. Image Processing*, vol. 3, pp. 367–381, July 1994.

[86] M. Barlaud, P. Solé, J. M. Moureaux, M. Antonini, and P. Gauthier, "Elliptical codebook for lattice vector quantization," in *Proc. ICASSP*, Minneapolis, MN, vol. 5, Apr. 1993, pp. 590–593.

[87] W. C. Powell and S. G. Wilson, "Lattice quantization in the wavelet domain," in *Proc. Math. Imaging: Wavelet Application Signal Image Processing '93*, vol. 2034, 1993, pp. 218–229.

[88] K. Ramchandran and M. Vetterli, "Best wavelet packet bases in a rate-distortion sense," *IEEE Trans. Image Processing*, vol. 2, no. 2, pp. 160–176, Apr. 1993.

[89] Z. Xiong, K. Ramchandran, M. T. Orchard, and K. Asai, "Wavelet packets-based image coding using joint space-frequency quantization," in *Proc. ICIP-94*, Austin, TX, vol. 3, Nov. 1994, pp. 324–328.

[90] P. Onno and C. Guillemot, "Wavelet packet coding with jointly optimized lattice vector quantization and data rate allocation," in *Proc. ICIP-94*, Austin, TX, vol. 3, Nov. 1994, pp. 329–333.

[91] L. Vandendorpe, "Human visual weighted quantization," *Ann. Télécommun.*, vol. 47, no. 7–8, pp. 282–292, 1992.

[92] B. Macq, "Weighted optimum bit allocations to orthogonal transforms for picture coding," *IEEE Trans. Selected Areas Commun.*, vol. 10, no. 5, pp. 875–883, June 1992.

[93] N. Moayeri, I. Daubechies, Q. Song, and H. S. Wang, "Wavelet transform image coding using trellis coded vector quantizers," in *Proc. ICASSP*, 1992, pp. 405–408.

[94] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. Image Processing*, vol. 1, no. 2, pp. 205–220, Apr. 1992.

[95] A. Singh and V. M. Bove, Jr., "Multidimensional quantizers for scalable video compression," *IEEE J. Selected Areas Commun.*, vol. 11, no. 1, pp. 36–45, Jan. 1993.

[96] M. Nomura, I. Furukawa, T. Fujui, and S. Ono, "Subband coding of super high definition images using entropy coded vector quantization," *IEICE Trans. Fundamentals Elec.*, vol. E75-A, no. 7, pp. 861–870, 1992.

[97] J.-R. Ohm, "Advanced packet-video coding based on layered VQ and

SBC techniques," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, no. 3, pp. 208–221, June 1993.

[98] C.-C. Lu and Y. H. Shin, "Interframe predictive coding of images using motion estimation and vector optimization," *IEEE Trans. Consumer Electron.*, vol. 38, no. 3, pp. 446–451, Aug. 1992.

[99] Y. Kato, Y. Yamada, H. Ohira, and T. Murakami, "A vector quantization of wavelet coefficients for super high definition image," in *Proc. Visual Commun. Image Processing '93*, vol. 2094, 1993, pp. 1021–1028.

[100] J. C. Feauveau, P. Mathieu, M. Barlaud, and M. Antonini, "Recursive biorthogonal wavelet transform for image coding," in *Proc. ICASSP*, Toronto, Canada, vol. 4, May 1991, pp. 2649–2652.

[101] M. Antonini, M. Barlaud, P. Mathieu, and J. C. Feauveau, "Multiscale image coding using the Kohonen neural network," in *Proc. SPIE Int. Soc. Opt. Eng.: Visual Commun. Image Processing*, Lausanne, Switzerland, vol. 1360, 1990, pp. 14–26.

[102] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using vector quantization in the wavelet transform domain," in *Proc. ICASSP*, Albuquerque, NM, USA, Apr. 1990, pp. 2297–2300.

[103] R. Baseri and V. J. Mathews, "Vector quantization of images using visual masking functions," in *Proc. ICASSP*, vol. 3, Mar. 1992, pp. 365–368.

[104] O. S. Haddadin, V. J. Mathews, and T. G. Stockham, Jr., "Subband vector quantization of images using hexagonal filter banks," in *Proc. Data Compression Conf.*, J. A. Storer and M. Cohn, Eds., Snowbird, UT, Mar. 1992, pp. 2–11.

[105] R. E. Van Dyck and S. A. Rajala, "Subband/VQ coding of color images using a separable diamond decomposition," *J. Visual Commun. Image Representation*, vol. 5, no. 3, pp. 205–220, Sept. 1994.

[106] Y.-Q. Zhang and W. Li, "A study of nonseparable subband filters for video coding," in *Proc. Visual Commun. Image Processing '92*, vol. 1818, 1992, pp. 233–240.

[107] J. N. Bradley and C. M. Brislawn, "Wavelet transform-vector quantization compression of supercomputer ocean models," in *Proc. 1993 IEEE Data Compression Conf. (DCC)*, J. A. Storer and M. Cohn, Eds., Snowbird, UT, USA, Mar. 1993, pp. 224–233.

[108] A. Cohen, I. Daubechies, and J. C. Feauveau, "Biorthogonal bases of compactly supported wavelets," Pub. TM 11217-900529-07, AT&T Bell Labs., Murray Hill, NJ, USA, 1990.

[109] J. N. Bradley and C. M. Brislawn, "Image compression by vector quantization of multiresolution decompositions," *Phys. D*, vol. 60, no. 1–4, pp. 245–258, Nov. 1992.

[110] A. N. Akansu and M. S. Kadur, "Subband coding of video with adaptive vector quantization," in *Proc. ICASSP*, Albuquerque, NM, USA, Apr. 1990, pp. 2109–2112.

[111] S.. Yao and R. J. Clarke, "Image sequence coding using adaptive vector quantisation in wavelet transform domain," *Electron. Lett.*, vol. 28, no. 17, pp. 1566–1568, Aug. 1992.

[112] H. Jafarkhani and N. Farvardin, "A scalable wavelet image coding scheme using multi-stage pruned tree-structured vector quantization," in *Proc. 1995 IEEE Int. Conf. Image Processing*, Washington DC, 1995.

[113] R. Rao and W. Pearlman, "Multirate vector quantization of image pyramids," in *Proc. ICASSP*, Toronto, Canada, May 1991, pp. 2657–2660.

[114] G. Furlan, C. Galand, E. Lançon, and J. Menez, "Sub-band coding of images using adaptive VQ and entropy coding," in *Proc. ICASSP*, Toronto, Canada, May 1991, pp. 2665–2668.

[115] C. Raimondo, C. Galand, E. Goubault, E. Lançon, and J. Menez, "Low bit-rate coder using hierarchical motion compensation and low-complexity vector quantization," in *Proc. ICASSP*, Toronto, Canada, May 1991, pp. 2717–2720.

[116] V. Ramasubramanian and K. K. Paliwal, "An efficient approximation-elimination algorithm for fast nearest-neighbor search based on a spherical distance coordinate formulation," in *Proc. Euro. Signal Processing Conf. (EUSIPCO 90)*, Barcelona, Sept. 1990, pp. 1323–1326.

[117] C. Stiller and O. Hirsch, "Subband coding of prediction error images using constrained-storage VQ," in *Proc. SPIE Int. Soc. Optical Eng.: Applications Digital Image Processing XV*, San Diego, CA, USA, vol. 1771, 1992, pp. 434–443.

[118] T. Denk, K. K. Parhi, and V. Cherkassky, "Combining neural networks and the wavelet transform for image compression," in *Proc. ICASSP*, Minneapolis, MN, USA, vol. 1, Apr. 1993, pp. 637–640.

[119] W. Li, "Vector transform and image coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 1, no. 4, pp. 297–307, Dec. 1991.

[120] W. Li and Y. Zhang, "A study of vector transform coding of subband-decomposed images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, no. 4, pp. 383–391, Aug. 1994.

[121] A. Pentland and B. Horowitz, "A practical approach to fractal-based image compression," in *IEEE Data Compression Conf.*, Mar. 1991, pp. 176–185.

[122] C. I. Podilchuk, N. S. Jayant, and P. Noll, "Sparse codebooks for the quantization of nondominant sub-bands in image coding," in *Proc. ICASSP*, Albuquerque, NM, USA, Apr. 1990, pp. 2101–2104.

[123] C. I. Podilchuk and N. Farvardin, "Perceptually based low bit rate video coding," in *Proc. ICASSP*, Toronto, Canada, May 1991, pp. 2837–2840.

[124] C. I. Podilchuk, N. S. Jayant, and N. Favardin, "Three-dimensional subband coding of video," *IEEE Trans. Image Processing*, vol. 4, no. 2, pp. 125–139, Feb. 1995.

[125] D. G. Sampson, E. A. B. da Silva, and M. Ghanbari, "Wavelet transform image coding using lattice vector quantisation," *Electron. Lett.*, vol. 30, no. 18, pp. 1477–1478, Sept. 1994.

[126] E. A. B. da Silva, D. G. Sampson, and M. Ghanbari, "Image coding using successive approximation wavelet vector quantisation," in *Proc. ICASSP*, Detroit, MI, 1995, pp. 2201–2204.

[127] D. G. Sampson, E. A. B. da Silva, and M. Ghanbari, "Wavelet vector quantisation scheme for image sequence coding at 64kbit/s," *Electron. Lett.*, vol. 31, no. 2, pp. 92–93, Jan. 1995.

[128] D. Taubman and A. Zakhor, "Multi-rate 3-D subband coding of video," *IEEE Trans. Image Processing,* Special Issue on Image Sequence Compression, vol. 3, no. 5, pp. 572–588, Sept. 1994.

[129] F. Kossentini, W. C. Chung, and M. J. T. Smith, "Subband image coding with jointly optimized quantizers," in *Proc. 1995 IEEE Int. Conf. Acoustics, Speech, Signal Processing*, May 1995, pp. 2221–2224.

[130] W. C. Chung, F. Kossentini, and M. J. T. Smith, "A new approach to scalable video coding," in *Proc. 1995 IEEE Data Compression Conf.*, Snowbird, UT, USA, Mar. 1995, pp. 381–390.

[131] P. C. Cosman, S. M. Perlmutter, and K. O. Perlmutter, "Tree-structured vector quantization with significance map for wavelet image coding," in *Proc. 1995 IEEE Data Compression Conf. (DCC)*, J. A. Storer and M. Cohn, Eds., Snowbird, UT, USA, Mar. 1995.

[132] P. H. Westerink, D. E. Boekee, J. Biemond, and J. W. Woods, "Subband coding of images using vector quantization," *IEEE Trans. Commun.*, vol. 36, pp. 713–719, June 1988.

[133] Y. H. Kim and J. W. Modestino, "Adaptive entropy coded subband coding of images," *IEEE Trans. Image Processing*, vol. 1, no. 1, pp. 31–48, 1992.

[134] R. F. Chang and Y. L. Huang, "Subband finite-state vector quantization," in *Proc. Visual Commun. Image Processing '94*, vol. 2308, 1994, pp. 177–188.

[135] Y. Huh, J. J. Hwang, C. K. Choi, R. de Queiroz, and K. R. Rao, "Classified wavelet transform coding of images using vector quantization," in *Proc. Visual Commun. Image Processing '94*, vol. 2308, 1994, pp. 207–217.

[136] Y. Huh, J. J. Hwang, and K. R. Rao, "Classified wavelet transform coding of images using two-channel conjugate vector quantization," in *Proc. ICIP-94*, Austin, TX, vol. 3, Nov. 1994, pp. 363–367.

[137] ——, "Block wavelet transform coding of images using classified vector quantization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, no. 1, pp. 63–67, Feb. 1995.

[138] X. Wang and S. Panchanathan, "Wavelet transform coding using NIVQ," in *Proc. Visual Commun. Image Processing '93*, vol. 2094, 1993, pp. 999–1009.

[139] A. Gersho, "Optimal nonlinear interpolative vector quantization," *IEEE Trans. Commun.*, vol. 38, no. 9-10, pp. 1285–1287, Sept. 1990.

[140] M. R. Banham, J. C. Brailean, and A. K. Katsaggelos, "A wavelet transform sequence coder using nonstationary displacement estimation," in *Proc. Visual Commun. Image Processing '92*, vol. 1818, 1992, pp. 210–221.

[141] M. R. Banham and B. J. Sullivan, "A wavelet transform image coding technique with a quadtree structure," in *Proc. ICASSP*, vol. 4, Mar. 1992, pp. 653–656.

[142] I. Moccagatta and M. Kunt, "VQ and cross-band prediction for color image coding," in *Proc. Picture Coding Symp. (PCS '94)*, Sacramento, CA, USA, Sept. 1994.

[143] ——, "An image coding scheme based on perceptually classified VQ for high compression ratios," in *Proc. 1995 IEEE Int. Symp. Acoustics, Speech, Signal Processing*, 1995, pp. 2487–2490.

[144] N. Mohsenian and N. Nasrabadi, "Edge-based subband VQ techniques for images and video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, no. 1, pp. 53–67, Feb. 1994.

[145] ——, "Subband coding of video using an edge-based vector quantization technique for compression of the upper bands," in *Proc. ICASSP*, 1992, pp. 233–236.

[146] J. Buhmann and H. Kühnel, "Complexity optimized vector quantization: A neural network approach," in *Proc. 1992 IEEE Data Compression Conf. (DCC)*, J. A. Storer and M. Cohn, Eds., Snowbird, UT, USA, Mar. 1992, pp. 12–21.

[147] J. Buhmann and H. Kühnel, "Vector quantization with complexity costs," *IEEE Trans. Inform. Theory*, vol. 39, no. 4, pp. 1133–1145, July 1988.

[148] B. Wegmann and C. Zetzsche, "Statistical dependence between orientation filter outputs used in an human vision based image code," in *Proc. Visual Commun. Image Processing '90*, vol. 1360, 1990, pp. 909–923.

[149] A. Nicoulin, M. Mattavelli, W. Li, and M. Kunt, "Subband image coding using jointly localized filter banks and entropy coding based on vector quantization," *Opt. Eng.*, vol. 32, no. 7, pp. 1438–1450, July 1993.

[150] B. Macq, S. Comes, J. Y. Mertès, and M. P. Queluz, "Signal-adapted transform coding of sequences," in *Proc. Visual Commun. Image Processing '93*, vol. 2094, 1993, pp. 1309–1320.

[151] C.-S. Kim, J. Bruder, M. J. T. Smith, and R. M. Mersereau, "Subband coding of color images using finite state vector quantization," in *Proc. ICASSP, 1988*, pp. 753–756.

[152] R. J. Safranek, J. D. Johnston, N. S. Jayant, and C. Podilchuk, "Perceptual coding of image signals," in *Proc. Twenty-fourth Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, Nov. 1990, pp. 346–350.

[153] D. Barba and J. Hanen, "The use of a human visual model in subband coding of color video signal with adaptive chrominance signal vector quantization," in *Proc. Visual Commun. Image Processing '91*, vol. 1605, 1991, pp. 408–419.

[154] R. E. Van Dyck and S. A. Rajala, "Subband/VQ coding in perceptually uniform color spaces," in *Proc. ICASSP*, vol. 3, Mar. 1992, pp. 237–240.

[155] ———, "Subband/VQ coding of color images with perceptually optimal bit allocation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, no. 1, pp. 68–82, Feb. 1994.

[156] C.-S. Kim, M. J. T. Smith, and R. M. Mersereau, "An improved SBC/VQ scheme for color image coding," in *Proc. ICASSP*, 1989, pp. 1941–1944.

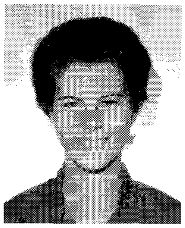
[157] M. H. Fadzil and T. J. Dennis, "Sample selection in subband vector quantization," in *Proc. ICASSP*, Albuquerque, NM, USA, Apr. 1990, pp. 2085–2088.

[158] C. I. E., "C. I. E. Colorimetry Committee proposal for study of color spaces," *J. Opt. Soc. Amer.*, vol. 64, no. 6, pp. 896–897, June 1974.

[159] R. E. Van Dyck, N. Moayeri, T. G. Marshall, Jr., and M. Chin, "Video coding using entropy-constrained trellis coded quantization," in *Proc. Applications Digital Image Processing XVII*, vol. 2298, 1994, pp. 119–130.

[160] I. Moccagatta and M. Kunt, "A pyramidal vector quantization approach to transform domain," in *Proc. EUSIPCO-92, Sixth Euro. Signal Processing Conf.*, Brussels, Aug. 1992, pp. 1365–1368.

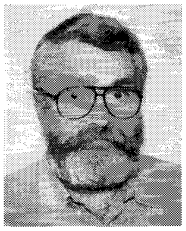
[161] R. M. Gray, S. J. Park, and B. Andrews, "Tiling shapes for image vector quantization," in *Proc. ComCon*, Victoria, Canada, Oct. 1991.



Pamela C. Cosman (M'93) received the B.S. degree in electrical engineering (with Honor) from the California Institute of Technology (Caltech), Pasadena, USA, in 1987, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, USA, in 1989 and 1993, respectively.

She was a lecturer and postdoctoral fellow at Stanford University (1993–1994) and a visiting assistant professor at the University of Minnesota (1994–1995). She is currently an assistant professor in the Department of Electrical and Computer Engineering at the University of California, San Diego, USA. Her research interests are in data compression and image processing.

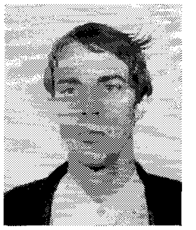
Dr. Cosman has received the Graduate and Postdoctoral Fellowships from the National Science Foundation (1987–1990 and 1994–1995), a Society of Women Engineers/General Electric Scholarship (1984–1987) and the Mabel Beckman Prize from Caltech (1987). She is a member of Tau Beta Pi and Sigma Xi. Her web page address is http://vision.ucsd.edu/~pcosman.



Robert M. Gray (S'68-M'69-SM'77-F'80) was born in San Diego, CA, USA, on November 1, 1943. He received the B.S. and M.S. degrees from the Massachusetts Institute of Technology (MIT), Cambridge, USA, in 1966 and the Ph.D. degree from the University of Southern California (USC), Los Angeles, USA, in 1969, all in electrical engineering.

Since 1969, he has been with Stanford University, Stanford, CA, USA, where he is currently a Professor and Vice Chair of the Department of Electrical Engineering. His research interests are the theory and design of data compression and classification systems, especially for medical images.

Dr. Gray was a member of the Board of Governors of the IEEE Information Theory Group (1974–1980, 1985–1988) as well as an Associate Editor (1977–1980) and Editor-in-Chief (1980–1983) of the IEEE Transactions on Information Theory. He is currently an Associate Editor of *Mathematics of Control, Signals, and Systems.* He was co-chair of the 1993 IEEE International Symposium on Information Theory. He is a member of the Image and Multidimensional Signal Processing Technical Committee of the IEEE Signal Processing Society. He has coauthored, with L. D. Davisson, *Random Processes* (Englewood Cliffs, NJ: Prentice Hall, 1986), with A. Gersho *Vector Quantization and Signal Compression* (Boston: Kluwer, 1992), and with J. W. Goodman *Fourier Transforms* (Boston: Kluwer). He is the author of *Probability, Random Processes, and Ergodic Properties* (New York: Springer-Verlag, 1988), *Source Coding Theory* (Boston: Kluwer, 1990), and *Entropy and Information Theory* (New York: Springer-Verlag, 1990). He was corecipient, with L. D. Davisson, of the 1976 IEEE Information Theory Group Paper Award and corecipient, with A. Buzo, A. H. Gray, and J. D. Markel, of the 1983 IEEE ASSP Senior Award. He was awarded an IEEE Centennial Medal in 1984 and the IEEE Signal Processing 1993 Society Award in 1994. He was elected Fellow of the Institute of Mathematical Statistics in 1992 and has held fellowships from the Japan Society for the Promotion of Science at the University of Osaka (1981), the Guggenheim Foundation at the University of Paris XI (1982), and NATO/Consiglio Nazionale delle Ricerche at the University of Naples (1990). During the spring of 1995, he was a Vinton Hayes Visiting Scholar at Harvard University, Boston, MA, USA. He is a member of Sigma Xi, Eta Kappa Nu, the AAAS, AMS, and the Societe des Ingénieurs et Scientifiques de France. He holds an Advanced Class Amateur Radio License (KB6XQ). His web page address is http://www-isl.stanford.edu/people/gray.



Martin Vetterli (F'95) received the Dipl. El.-Ing. degree from ETH Zurich, Switzerland, in 1981, the M.S. degree from Stanford University, Stanford, CA, USA, in 1982, and the Doctorat ès Science degree from EPF Lausanne, Switzerland, in 1986.

He was a Research Assistant at Stanford and EPFL and has worked for Siemens and AT&T Bell Laboratories. In 1986, he joined Columbia University, New York, NY, USAS, where he was last an Associate Professor of Electrical Engineering and co-director of the Image and Advanced Television Laboratory. In 1993, he joined the University of California at Berkeley, where he is currently a Professor in the Department of Electrical Engineering and Computer Sciences. His research interests include wavelets, multirate signal processing, computational complexity, signal processing for telecommunications, and digital video processing and compression.

Dr. Vetterli is a member of SIAM and of the editorial boards of *Signal Processing, Image Communication, Annals of Telecommunications, Applied and Computational Harmonic Analysis,* and *The Journal of Fourier Analysis and Applications.* He received the Best Paper Award from EURASIP in 1984 for his paper on multidimensional subband coding, the Research Prize from the Brown Bovery Corporation (Switzerland) in 1986 for his thesis, and the IEEE Signal Processing Society's 1991 Senior Award for a 1989 Transactions paper with D. LeGall. He was a plenary speaker at the *1992 IEEE ICASSP* in San Francisco, CA, USA, and is the co-author, with J. Kovacevic, of *Wavelets and Subband Coding* (Englewood Cliffs, NJ: Prentice-Hall, 1995).