1 SUBBAND IMAGE COMPRESSION

Aria Nosratinia[1], Geoffrey Davis[2],
Zixiang Xiong[3], and Rajesh Rajagopalan[4]

[1]Dept of Electrical and Computer Engineering, Rice University, Houston, TX 77005
[2]Math Department, Dartmouth College, Hanover, NH 03755
[3]Dept. of Electrical Engineering, University of Hawaii, Honolulu, HI 96822
[4]Lucent Technologies, Murray Hill, NJ 07974

Abstract: This chapter presents an overview of subband/wavelet image compression. Shannon showed that optimality in compression can only be achieved with Vector Quantizers (VQ). There are practical difficulties associated with VQ, however, that motivate transform coding. In particular, reverse waterfilling arguments motivate subband coding. Using a simplified model of a subband coder, we explore key design issues. The role of smoothness and compact support of the basis elements in compression performance are addressed. We then look at the evolution of practical subband image coders. We present the rudiments of three generations of subband coders, which have introduced increasing degrees of sophistication and performance in image compression: The first generation attracted much attention and interest by introducing the zerotree concept. The second generation used adaptive space-frequency and rate-distortion optimized techniques. These first two generations focused largely on inter-band dependencies. The third generation includes exploitation of intra-band dependencies, utilizing trellis-coded quantization and estimation-based methods. We conclude by a summary and discussion of future trends in subband image compression.

1.1 INTRODUCTION

Digital imaging has had an enormous impact on industrial applications and scientific projects. It is no surprise that image coding has been a subject of great commercial interest. The JPEG image coding standard has enjoyed widespread acceptance, and the industry continues to explore issues in its implementation.

In addition to being a topic of practical importance, the problems studied in image coding are also of considerable theoretical interest. The problems draw upon and have inspired work in information theory, applied harmonic analysis, and signal processing. This chapter presents an overview of subband image coding, arguably one of the most fruitful and successful directions in image coding.

1.1.1 Image Compression

An image is a positive function on a plane. The value of this function at each point specifies the luminance or brightness of the picture at that point.[1] Digital images are sampled versions of such functions, where the value of the function is specified only at discrete locations on the image plane, known as *pixels*. The value of the luminance at each pixel is represented to a pre-defined precision M. Eight bits of precision for luminance is common in imaging applications. The eight-bit precision is motivated by both the existing computer memory structures (1 byte = 8 bits) as well as the dynamic range of the human eye.

The prevalent custom is that the samples (pixels) reside on a rectangular lattice which we will assume for convenience to be $N \times N$. The brightness value at each pixel is a number between 0 and $2^M - 1$. The simplest binary representation of such an image is a list of the brightness values at each pixel, a list containing N^2M bits. Our standard image example in this paper is a square image with 512 pixels on a side. Each pixel value ranges from 0 to 255, so this canonical representation requires $512^2 \times 8 = 2,097,152$ bits.

Image coding consists of mapping images to strings of binary digits. A good image coder is one that produces binary strings whose lengths are on average much smaller than the original canonical representation of the image. In many imaging applications, exact reproduction of the image bits is not necessary. In this case, one can perturb the image slightly to obtain a shorter representation. If this perturbation is much smaller than the blurring and noise introduced in the formation of the image in the first place, there is no point in using the more accurate representation. Such a coding procedure, where perturbations reduce storage requirements, is known as *lossy coding*. The goal of lossy coding is to reproduce a given image with minimum distortion, given some constraint on the total number of bits in the coded representation.

[1] Color images are a generalization of this concept, and are represented by a three-dimensional vector function on a plane. In this paper, we do not explicitly treat color images. However, most of the results can be directly extended to color images.

But why can images be compressed on average? Suppose for example that we seek to efficiently store any image that has ever been seen by a human being. In principle, we can enumerate all images that have ever been seen and represent each image by its associated index. We generously assume that some 50 billion humans have walked the earth, that each person can distinguish on the order of 100 images per second, and that people live an average of 100 years. Combining these figures, we estimate that humans have seen some 1.6×10^{22} images, an enormous number. However, $1.6 \times 10^{22} \approx 2^{73}$, which means that the entire collective human visual experience can be represented with a mere 10 bytes (73 bits, to be precise)!

This collection includes any image that a modern human eye has ever seen, including artwork, medical images, and so on, yet the collection can be conceptually represented with a small number of bits. The remaining vast majority of the $2^{512\times512\times8} \approx 10^{600,000}$ possible images in the canonical representation are not of general interest, because they contain little or no structure, and are noise-like.

While the above conceptual exercise is intriguing, it is also entirely impractical. Indexing and retrieval from a set of size 10^{22} is completely out of the question. However, we can see from the example the two main properties that image coders exploit. First, only a small fraction of the possible images in the canonical representation are likely to be of interest. *Entropy coding* can yield a much shorter image representation on average by using short code words for likely images and longer code words for less likely images.[2] Second, in our initial image gathering procedure we sample a continuum of possible images to form a discrete set. The reason we can do so is that most of the images that are left out are visually indistinguishable from images in our set. We can gain additional reductions in stored image size by discretizing our database of images more coarsely, a process called *quantization*. By mapping visually indistinguishable images to the same code, we reduce the number of code words needed to encode images, at the price of a small amount of distortion.

1.1.2 Outline of the Chapter

It is possible to quantize each pixel separately, a process known as *scalar quantization*. Quantizing a group of pixels together is known as *vector quantization*, or VQ. Vector quantization can, in principle, capture the maximum compression that is theoretically possible. In Section 1.2 we review the basics of quantization, vector quantization, and the mechanisms of gain in VQ.

VQ is a very powerful theoretical paradigm, and can asymptotically achieve optimality. But the computational cost and delay also grow exponentially with dimensionality, limiting the practicality of VQ. Due to these and other difficulties, most practical coding algorithms have turned to *transform coding* instead

[2]For example, mapping the ubiquitous test image of Lena Sjööblom (see Figure 1.12) to a one-bit codeword would greatly compress the image coding literature.

of high-dimensional VQ. Transform coding usually consists of scalar quantization in conjunction with a linear transform. This method captures much of the VQ gain, with only a fraction of the effort. In Section 1.3, we present the fundamentals of transform coding. We use a second-order model to motivate the use of transform coding.

The success of transform coding depends on how well the basis functions of the transform represent the features of the signal. At present, one of the most successful representations is the *subband/wavelet transform*. A complete derivation of fundamental results in subband signal analysis is beyond the scope of this chapter, and the reader is referred to excellent existing references such as [1, 2]. The present discussion focuses on compression aspects of subband transforms.

Section 1.4 outlines the key issues in subband coder design, from a general transform coding point of view. However, the general transform coding theory is based only on second-order properties of a random model of the signal. While subband coders fit into the general transform coding framework, they also go beyond it. Because of their nice temporal properties, subband decompositions can capture redundancies beyond general transform coders. We describe these extensions in Section 1.5, and show how they have motivated some of the most recent coders, which we describe in Sections 1.6, 1.7 and 1.8. We conclude by a summary and discussion of future directions.

1.2 QUANTIZATION

At the heart of image compression is the idea of quantization and approximation. While the images of interest for compression are almost always in a digital format, it is instructive and more mathematically elegant to treat the pixel luminances as being continuously valued. This assumption is not far from the truth if the original pixel values are represented with a large number of levels.

The role of quantization is to represent this continuum of values with a finite — preferably small — amount of information. Obviously this is not possible without some loss. The quantizer is a function whose set of output values are discrete and usually finite (see Figure 1.1). Good quantizers are those that represent the signal with a minimum distortion.

Figure 1.1 also indicates a useful view of quantizers as concatenation of two mappings. The first map, the *encoder*, takes partitions of the x-axis to the set of integers $\{-2, -1, 0, 1, 2\}$. The second, the *decoder*, takes integers to a set of output values $\{\hat{x}_k\}$. We need to define a measure of distortion in order to characterize "good" quantizers. We need to be able to approximate any possible value of x with an output value $\hat{x}_k$. Our goal is to minimize the distortion on average, over all values of x. For this, we need a probabilistic model for the signal values. The strategy is to have few or no reproduction points in locations at which the probability of the signal is negligible, whereas at highly probable signal values, more reproduction points need to be specified. While improbable values of x can still happen — and will be costly — this strategy pays off *on*

average. This is the underlying principle behind all signal compression, and will be used over and over again in different guises.

The same concepts apply to the case where the input signal is not a scalar, but a vector. In that case, the quantizer is known as a Vector Quantizer (VQ).

1.2.1 Vector Quantization

Vector quantization (VQ) is the generalization of scalar quantization to the case of a vector. The basic structure of a VQ is essentially the same as scalar quantization, and consists of an encoder and a decoder. The encoder determines a partitioning of the input vector space and to each partition assigns an index, known as a *codeword*. The set of all codewords is known as a *codebook*. The decoder maps the each index to a reproduction vector. Combined, the encoder and decoder map partitions of the space to a discrete set of vectors.

Vector Quantization is a very important concept in compression: In 1959 Shannon [3] delineated fundamental limitations of compression systems through his "Source coding theorem with a fidelity criterion." While this is not a constructive result, it does indicate, loosely speaking, that fully effective compression can only be achieved when input data samples are encoded in blocks of increasing length, i.e. in large vectors.

Optimal vector quantizers are not known in closed form except in a few trivial cases. However, two optimality conditions are known for VQ (and for scalar quantization as a special case) which lead to a practical algorithm for the design of quantizers. These conditions were discovered independently by Lloyd [4, 5] and Max [6] for scalar quantization, and were extended to VQ by Linde, Buzo, and Gray [7]. An example of cell shapes for a two-dimensional optimal quantizer is shown in Figure 1.2. We state the result here and refer the reader to [8] for proof.

Let $p_{\mathbf{X}}(\mathbf{x})$ be the probability density function for the random variable $\mathbf{X}$ we wish to quantize. Let $D(\mathbf{x}, \mathbf{y})$ be an appropriate distortion measure. Like scalar quantizers, vector quantizers are characterized by two operations, an encoder

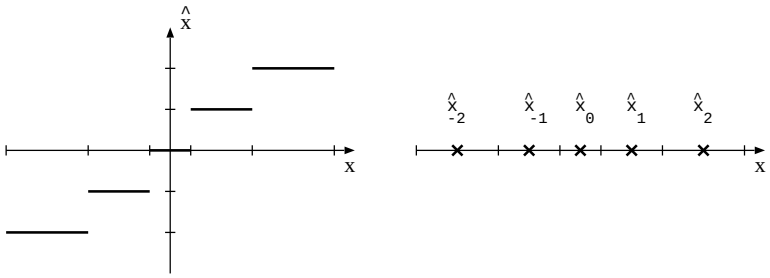

Figure 1.1 (Left) Quantizer as a function whose output values are discrete. (Right) because the output values are discrete, a quantizer can be more simply represented only on one axis.

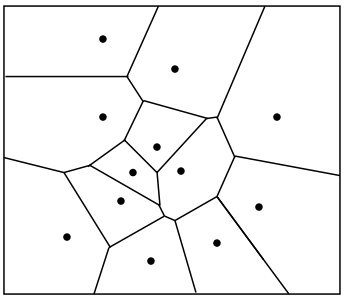

Figure 1.2 A Voronoi Diagram

and a decoder. The encoder is defined by a partition of the range of $\mathbf{X}$ into sets $\mathcal{P}_k$. All realizations of $\mathbf{X}$ that lie in $\mathcal{P}_k$ are encoded to k and decoded to $\hat{\mathbf{x}}_k$. The decoder is defined by specifying the reproduction value $\hat{\mathbf{x}}_k$ for each partition $\mathcal{P}_k$.

A quantizer that minimizes the average distortion D must satisfy the following conditions:

1. *Nearest neighbor condition:* Given a set of reconstruction values $\{\hat{\mathbf{x}}_k\}$, the optimal partition of the values of $\mathbf{X}$ into sets $\mathcal{P}_k$ is the one for which each value $\mathbf{x}$ is mapped by the encoding and decoding process to the nearest reconstruction value. Thus,

$$\mathcal{P}_k = \{\mathbf{x} : D(\mathbf{x}, \hat{\mathbf{x}}_k) \leq D(\mathbf{x}, \hat{\mathbf{x}}_j) \text{ for } j \neq k\}. \tag{1.1}$$

2. *Centroid condition:* Given a partition of the range of $\mathbf{X}$ into sets $\mathcal{P}_k$, the optimal reconstruction values values $\hat{\mathbf{x}}_k$ are the generalized centroids of the sets $\mathcal{P}_k$. They satisfy

$$\hat{\mathbf{x}}_k = \arg\min \int_{\mathcal{P}_k} p_{\mathbf{X}}(\mathbf{z}) D(\mathbf{z}, \hat{\mathbf{x}}_k) d\mathbf{z}. \tag{1.2}$$

   With the squared error distortion, the generalized centroid corresponds to the $p_{\mathbf{X}}(\mathbf{x})$-weighted centroid.

1.2.2 Limitations of VQ

Although vector quantization is a very powerful tool, the computational and storage requirements become prohibitive as the dimensionality of the vectors increase. The complexity of VQ has motivated a wide variety of constrained VQ methods. Among the most prominent are tree structured VQ, shape-gain VQ, classified VQ, multistage VQ, lattice VQ, and hierarchical VQ [8].

There is another important consideration that limits the practical use of VQ in its most general form: the design of the optimal quantizer requires knowledge of the underlying probability density function for the space of images. While we

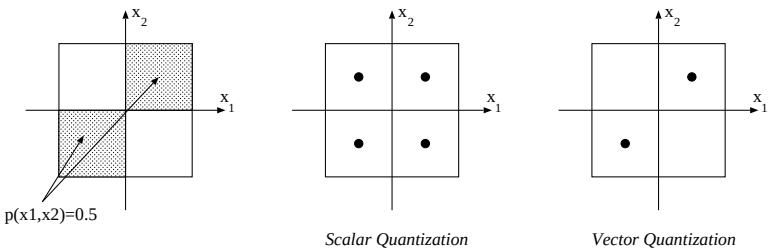

Figure 1.3 Leftmost figure shows a probability density for a two-dimensional vector $\mathbf{X}$. The realizations of $\mathbf{X}$ are uniformly distributed in the shaded areas. Center figure shows the four reconstruction values for an optimal scalar quantizer for $\mathbf{X}$ with expected squared error $\frac{1}{12}$. The figure on the right shows the two reconstruction values for an optimal vector quantizer for $\mathbf{X}$ with the same expected error. The vector quantizer requires 0.5 bits per sample, while the scalar quantizer requires 1 bit per sample.

may claim empirical knowledge of lower order joint probability distributions, the same is not true of higher orders. A training set is drawn from the distribution we are trying to quantize, and is used to drive the algorithm that generates the quantizer. As the dimensionality of the model is increased, the amount of data available to estimate the density in each bin of the model decreases, and so does the reliability of the p.d.f. estimate.[3] The issue is commonly known as "the curse of dimensionality."

Instead of accommodating the complexity of VQ, many compression systems opt to move away from it and employ techniques that allow them to use sample-wise or scalar quantization more effectively. To design more effective scalar quantization systems, however, one needs to know the source of the compression efficiency of the VQ. Then one can try to capture as much of that efficiency as possible, in the context of a scalar quantization system.

1.2.3 Why VQ Works

The source of the compression efficiency of VQ is threefold: (a) exploiting correlation redundancy, (b) sphere covering and density shaping, and (c) exploiting fractional bitrates.

Correlation Redundancy. The greatest benefit of jointly quantizing random variables is that we can exploit the dependencies between them. Figure 1.3 shows a two-dimensional vector $\mathbf{X} = (X_1, X_2)$ that is distributed uniformly over the squares $[0,1]\times[0,1]$ and $[-1,0]\times[-1,0]$. The marginal densities for X_1 and

[3]Most existing techniques do not estimate the p.d.f. to use it for quantization, but rather use the data directly to generate the quantizer. However, the reliability problem is best pictured by the p.d.f. estimation exercise. The effect remains the same with the so-called direct or data-driven methods.

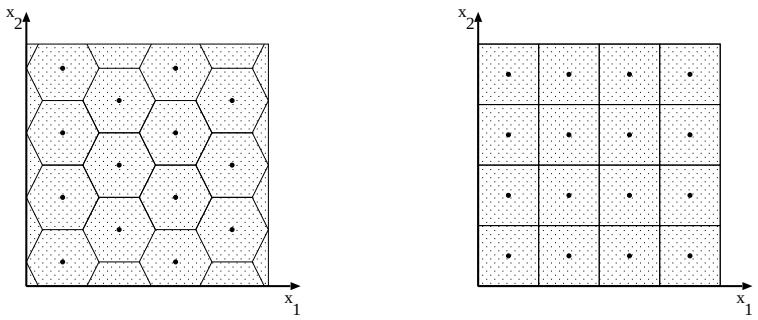

Figure 1.4 Tiling of the two-dimensional plane. The hexagonal tiling is more efficient, leading to a better rate-distortion.

X_2 are both uniform on $[-1, 1]$. We now hold the expected distortion fixed and compare the cost of encoding X_1 and X_2 as a vector, to the cost of encoding these variables separately. For an expected squared error of $\frac{1}{12}$, the optimal scalar quantizer for both X_1 and X_2 is the one that partitions the interval $[-1, 1]$ into the subintervals $[-1, 0)$ and $[0, 1]$. The cost per symbol is 1 bit, for a total of 2 bits for $\mathbf{X}$. The optimal vector quantizer with the same average distortion has cells that divides the square $[-1, 1] \times [-1, 1]$ in half along the line $y = -x$. The reconstruction values for these two cells are $\hat{\mathbf{x}}_a = (-\frac{1}{2}, -\frac{1}{2})$ and $\hat{\mathbf{x}}_b = (\frac{1}{2}, \frac{1}{2})$. The total cost per vector $\mathbf{X}$ is just 1 bit, only half that of the scalar case.

Because scalar quantizers are limited to using separable partitions, they cannot take advantage of dependencies between random variables. This is a serious limitation, but we can overcome it in part through a preprocessing step consisting of a linear transform.

Sphere Covering and Density Shaping. Even if random components of a vector are *independent*, there is some gain in quantizing them jointly, rather than independently. This may at first seem surprising, but is universally true and is due to the geometries of multidimensional spaces. We demonstrate by an example.

Assume we intend to quantize two uniformly distributed, independent random variables $\mathbf{X}_1$ and $\mathbf{X}_2$. One may quantize them independently through two scalar quantizers, leading to a rectangular tiling of the $x_1 - x_2$ plane. Figure 1.4 shows this, as well as a second quantization strategy with hexagonal tiling. Assuming that these rectangles and hexagons have the same area, and hence the same rate (we disregard boundary effects), the squared error from the hexagonal partition is 3.8% lower than that of the square partition due to the extra error contributed by the corners of the rectangles.

In other words, one needs to cover the surface with shapes that have maximal ratio of area to moment-of-inertia. It is known that the best two-dimensional

shape in that respect is the circle. It has also been shown that the best tiling of the 2-D plane in that respect is achieved by the hexagon (so our example is in fact optimal).

Generally, in n-dimensional spaces, the performance of vector quantizers is determined in part by how closely we can approximate spheres with n-dimensional convex polytopes [9]. When we quantize vector components separately using scalar quantizers, the resulting Voronoi cells are all rectangular prisms, which only poorly approximate spheres. VQ makes it possible to use geometrically more efficient cell shapes. The benefits of improved spherical approximations increase in higher dimensions. For example, in 100 dimensions, the optimal vector quantizer for uniform densities has an error of roughly 0.69 times that of the optimal scalar quantizer for uniform densities, corresponding to a PSNR gain of 1.6 dB [9].

This problem is closely related to the well-studied problem of sphere covering in lattices. The problem remains largely unsolved, except for the uniform density at dimensions 2, 3, 8, and 24. Another noteworthy result is due to Zador [10], which gives asymptotic cell densities for high-resolution quantization.

Fractional Bitrates. In scalar quantization, each input sample is represented by a separate codeword. Therefore, the minimum bitrate achievable is one bit per sample, because our symbols cannot be any shorter than one bit. Since each symbol can only have an integer number of bits, one can generate fractional bitrates per sample by coding multiple samples together, as done in vector quantization. A vector quantizer coding N-dimensional vectors using a K-member codebook can achieve a rate of $(\log_2 K)/N$ bits per sample. For example, in Figure 1.3 scalar quantization cannot have a rate lower than one bit per sample, while vector quantization achieves the same distortion with 0.5 bits per sample.

The problem with fractional bitrates is especially acute when one symbol has very high probability and hence requires a very short code length. For example, the zero symbol is very commonly used when coding the high-frequency portions of subband-transformed images. The only way of obtaining the benefit of fractional bitrates with scalar quantization is to jointly re-process the codewords after quantization. Useful techniques to perform this task include arithmetic coding, run-length coding (as in JPEG), and zerotree coding.

Finally, the three mechanisms of gain noted above are not always separable and independent of each other, and processing aimed at capture one form of gain one may capture others as well. For example, run-length coding and zerotree coding are techniques that enable the attainment of fractional bitrates as well as the partial capture of correlation redundancy.

1.3 TRANSFORM CODING

The advantage of VQ over scalar quantization is primarily due to VQ's ability to exploit dependencies between samples. Direct scalar quantization of the

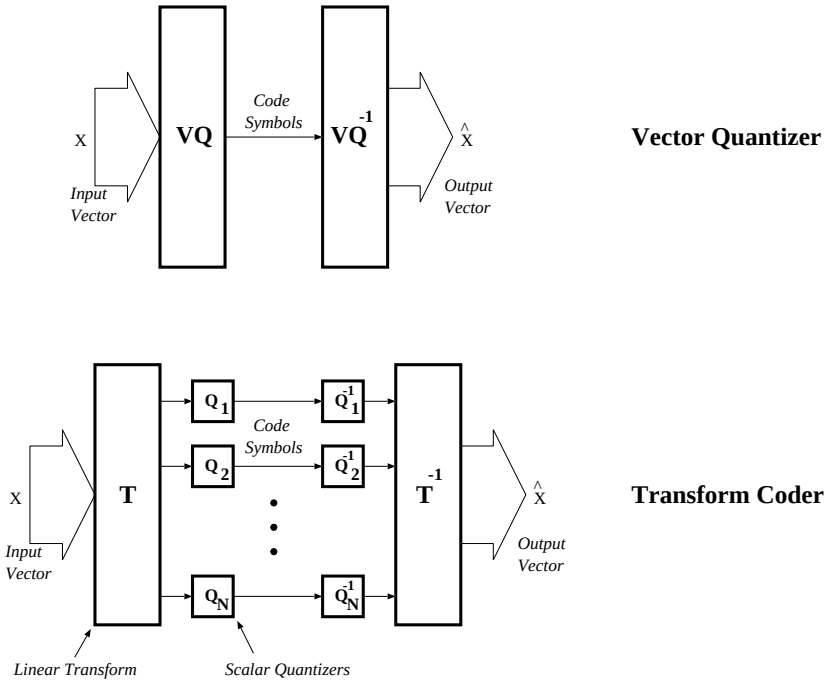

Figure 1.5 Transform coding simplifies the quantization process by applying a linear transform.

samples does not capture this redundancy, and therefore suffers. However, we have seen that VQ presents severe practical difficulties, so the usage of scalar quantization is highly desirable. Transform coding is one mechanism by which we can capture the correlation redundancy, while using scalar quantization (Figure 1.5).

Transform coding does not capture the geometrical "packing redundancy," but this is usually a much smaller factor than the correlation redundancy. Scalar quantization also does not address fractional bitrates by itself, but other post-quantization operations can capture the advantage of fractional bitrates with manageable complexity (e.g. zerotrees, run-length coding, arithmetic coding).

To illustrate the exploitation of correlation redundancies by transform coding, we consider a toy image model. Images in our model consist of two pixels, one on the left and one on the right. We assume that these images are realizations of a two-dimensional random vector $\mathbf{X} = (X_1, X_2)$ for which X_1 and X_2 are identically distributed and jointly Gaussian. The identically distributed assumption is a reasonable one, since there is no *a priori* reason that pixels on the left and on the right should be any different. We know empirically that adjacent image pixels are highly correlated, so let us assume that the autocorrelation

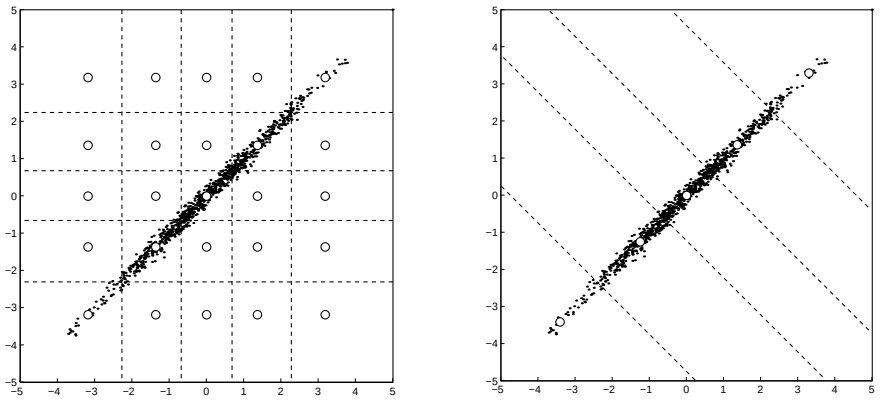

Figure 1.6 Left: Correlated Gaussians of our image model quantized with optimal scalar quantization. Many reproduction values (shown as white dots) are wasted. Right: Decorrelation by rotating the coordinate axes. The new axes are parallel and perpendicular to the major axis of the cloud. Scalar quantization is now much more efficient.

matrix for these pixels is

$$E[\mathbf{X}\mathbf{X}^T] = \begin{bmatrix} 1 & 0.9 \\ 0.9 & 1 \end{bmatrix} \tag{1.3}$$

By symmetry, X_1 and X_2 will have identical quantizers. The Voronoi cells for this scalar quantization are shown on the left in Figure 1.6. The figure clearly shows the inefficiency of scalar quantization: most of the probability mass is concentrated in just five cells. Thus a significant fraction of the bits used to code the bins are spent distinguishing between cells of very low probability. This scalar quantization scheme does not take advantage of the coupling between X_1 and X_2.

We can remove the correlation between X_1 and X_2 by applying a rotation matrix. The result is a transformed vector $\mathbf{Y}$ given by

$$\mathbf{Y} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \tag{1.4}$$

This rotation does not remove any of the variability in the data. Instead it packs that variability into the variable Y_1. The new variables Y_1 and Y_2 are independent, zero-mean Gaussian random variables with variances 1.9 and 0.1, respectively. By quantizing Y_1 finely and Y_2 coarsely we obtain a lower average error than by quantizing X_1 and X_2 equally. In the remainder of this section we will describe general procedures for finding appropriate redundancy-removing transforms, and for optimizing related quantization schemes.

1.3.1 The Karhunen-Loève Transform

The previous simple example shows that removing correlations can lead to better compression. One can remove the correlation between a group of random variables using an orthogonal linear transform called the Karhunen-Loève transform (KLT), also known as the Hotelling transform.

Let $\mathbf{X}$ be a random vector that we assume has zero-mean and autocorrelation matrix $\mathbf{R}_X$. The Karhunen-Loève transform is the matrix $\mathbf{A}$ that will make the components of $\mathbf{Y} = \mathbf{AX}$ uncorrelated. It can be easily verified that such a transform matrix $\mathbf{A}$ can be constructed from the eigenvectors of $\mathbf{R}_X$, the autocorrelation matrix of $\mathbf{X}$. Without loss of generality, the rows of $\mathbf{A}$ are ordered so that $\mathbf{R_Y} = \mathbf{diag}(\lambda_\mathbf{0}, \lambda_\mathbf{1}, \ldots, \lambda_{\mathbf{N-1}})$ where $\lambda_0 \geq \lambda_1 \geq ... \geq \lambda_{N-1} \geq 0$.

This transform is optimal among all block transforms, in the sense described by the two theorems below (see [11] for proofs). The first theorem states that the KLT is optimal for mean-squares approximation over a large class of random vectors.

Theorem 1 *Suppose that we truncate a transformed random vector* $\mathbf{AX}$*, keeping* m *out of the* N *coefficients and setting the rest to zero, then among all linear transforms, the Karhunen-Loève transform provides the best approximation in the mean square sense to the original vector.*

The KLT is also optimal among block transforms in the rate-distortion sense, but only when the input is a Gaussian vector and for high-resolution quantization. Optimality is achieved with a quantization strategy where the quantization noise from all transform coefficients are equal [11].

Theorem 2 *For a zero-mean, jointly Gaussian random vector, and for high-resolution quantization, among all block transforms, the Karhunen-Loève transform minimizes the distortion at a given rate.*

We emphasize that the KLT is optimal only in the context of block transforms, and partitioning an image into blocks leads to a reduction of performance. It can be shown [12] that subband transforms, which are not block-based, can provide better energy compaction properties than a block-based KLT. In the next section we motivate the use of subband transforms in coding applications using reverse waterfilling arguments.

1.3.2 Reverse Waterfilling and Subband Transforms

The limitations of block-based Karhunen-Loève transforms result from the blocking of the source. We can eliminate blocking considerations by restricting our attention to a stationary source and taking the block size to infinity. Stationary random processes have Toeplitz autocorrelation matrices. The eigenvectors of a circulant matrix are known to be complex exponentials, thus a large Toeplitz matrix with sufficiently decaying off-diagonal elements will have a diagonalizing transform close to the Discrete Fourier Transform (DFT). In

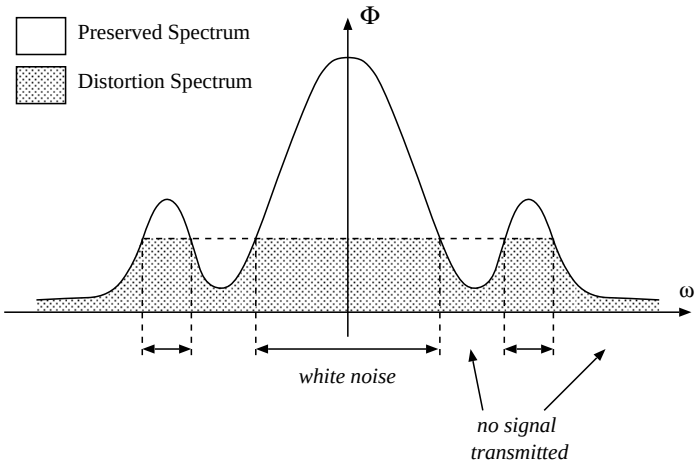

Figure 1.7 Reverse water filling of the spectrum for the rate-distortion function of a Gaussian source with memory.

other words, with sufficiently large block sizes, the KLT of a stationary process resembles the Fourier transform. In particular, one can make more precise statements about the KL *transform coefficients*. It has been shown [13] that in the limiting case when the block size goes to infinity, the distribution of KL transform coefficients approaches that of the Fourier spectrum of the autocorrelation.

The optimality of KLT for block-based processing of Gaussian processes and the limiting results in [13] suggest that, when taking block sizes to infinity, power spectral density (psd) is the appropriate vehicle for bit allocation purposes. Similarly to the case of finite-dimensional KLT, our bit allocation procedure consists of discarding very low-energy components of psd, and quantizing the remaining components such that each coefficient contributes an equal amount of distortion [11]. This concept is known as *reverse waterfilling*.

Reverse waterfilling can also be directly derived from a rate-distortion perspective. Unlike the "limiting KLT" argument described above, this explanation is not bound to high-resolution quantization and is therefore more general. Consider a Gaussian source with memory (i.e. correlated) with power spectral density $\Phi_X(\omega)$. The rate-distortion function can be expressed parametrically [14]

$$D(\theta) = \frac{1}{2\pi^2} \int_{\omega} \min(\theta, \Phi_X(\omega)) d\omega \tag{1.5}$$

$$R(\theta) = \frac{1}{4\pi^2} \int_{\omega} \max\left(0, \log(\frac{\Phi_X(\omega)}{\theta})\right) d\omega \tag{1.6}$$

R and D are the rate and distortion pairs predicted by the Shannon limit, parameterized by θ. The goal is to design a quantization scheme that approach

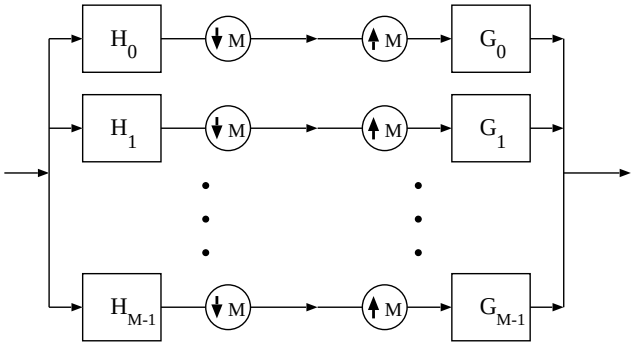

Figure 1.8 Filter bank

this theoretical rate-distortion limit. Our strategy is: at frequencies where signal power is less than θ, it is not worthwhile to spend any bits, therefore all the signal is thrown away (signal power = noise power). At frequencies where signal power is greater than θ, enough bitrate is assigned so that the noise power is exactly θ, and signal power over and above θ is preserved. Reverse waterfilling is illustrated in Figure 1.7.

In reverse waterfilling, each frequency component is quantized with a separate quantizer, reflecting the bit allocation appropriate for that particular component. For the Gaussian source, each frequency component is a Gaussian with variance given by the power spectrum. The process of quantizing these frequencies can be simplified by noting that frequencies with the same power density use the same quantizer. As a result, our task is simply to divide the spectrum into a partition of white segments, and to assign a quantizer to each segment. We achieve an optimal tradeoff between rate and distortion by this procedure for piecewise-constant power spectra. For other reasonably smooth power spectra, we can approach optimality by partitioning the spectrum into segments that are approximately white and quantizing each segment individually.

Thus, removing blocking constraints lead to reverse waterfilling arguments which in turn motivate separation of the source into frequency bands. This separation is achieved by subband transforms, which are implemented by filter banks.

A subband transformer is a multi-rate digital signal processing system. As shown in Figure 1.8, it consists of two sets of filter banks, along with decimators and interpolators. On the left side of the figure we have the forward stage of the subband transform. The signal is sent through the input of the first set of filters, known as the *analysis filter bank*. The output of these filters is passed through decimators, which retain only one out of every M samples. The right hand side of the figure is the inverse stage of the transform. The filtered and decimated signal is first passed through a set of *interpolators*. Next it is passed through the *synthesis filter bank*. Finally, the components are recombined.

The combination of decimation and interpolation has the effect of zeroing out all but one out of M samples of the filtered signal. Under certain conditions, the original signal can be reconstructed exactly from this decimated M-band representation. The ideas leading to the perfect reconstruction conditions were discovered in stages by a number of investigators, including Croisier *et al.* [15], Vaidyanathan [16], Smith and Barnwell [17, 18] and Vetterli [19, 20]. For a detailed presentation of these developments, we refer the reader to the comprehensive texts by Vaidyanathan [2] and Vetterli and Kovačević [1].

1.3.3 Hierarchical Subbands, Wavelets, and Smoothness

A subset of subband transforms has been very successful in image compression applications; we refer to hierarchical subbands and in particular wavelet transforms. In this section we discuss reasons for the suitability of these transforms for image coding.

The waterfilling algorithm motivates a frequency domain approach to quantization and bit allocation. It is generally accepted that images of interest, considered as a whole, have power spectra that are stronger at lower frequencies. In particular, many use the exponentially decaying model for the tail of the power spectrum given by

$$S_X(\omega) = e^{-\alpha|\omega|} \; . \tag{1.7}$$

We can now apply the waterfilling algorithm. Since the spectral model is not piecewise constant, we need to break it up in such a way that the spectrum is approximately constant in each segment. Applying a minimax criterion for the approximation yields a logarithmically distributed set of frequency bands. As we go from low frequency bands to high, the length of each successive band increases by a constant factor that is greater than 1. This in turn motivates a hierarchical structure for the subband decomposition of the signal (see Figure 1.9).

Hierarchical decompositions possess a number of additional attractive features. One of the most important is that they provide a measure of *scale invariance* in the transform. Consider that a shift of the location of the viewer results (roughly) in a translation and rescaling of the perceived image. We have no *a priori* reason to expect any particular viewer location; as a result, natural images possess no favored translates or scalings. Subband transforms are invariant under translates by K pixels (where K depends on the transform) since they are formed by convolution and downsampling. Hierarchical transforms add an additional degree of scale invariance. The result is a family of coding algorithms that work well with images at a wide variety of scales.

A second advantage of hierarchical subband decompositions is that they provide a convenient tree structure for the coded data. This turns out to be very important for taking advantage of remaining correlations in the signal (because image pixels, unlike our model, are not generally jointly Gaussian). We will see that zerotree coders use this structure with great efficiency.

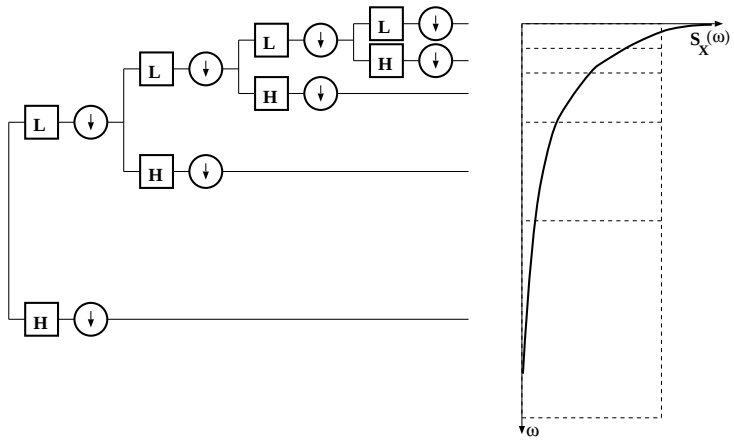

Figure 1.9 Exponential decay of power density motivates a logarithmic frequency division, leading to a hierarchical subband structure.

A third advantage of hierarchical decompositions is that they leverage a considerable body of work on wavelets. The discrete wavelet transform is functionally equivalent to a hierarchical subband transform, and each framework brings to bear an important perspective on the problem of designing effective transforms. As we have seen, the subband perspective is motivated by frequency-domain arguments about optimal compression of stationary Gaussian random processes. The wavelet perspective, in contrast, emphasizes frequency as well as spatial considerations. This spatial emphasis is particularly useful for addressing nonstationary behavior in images, as we will see in the discussion of coders below.

Both the wavelet and subband perspectives yield useful design criteria for constructing filters. The subband framework emphasizes coding gain, while the wavelet framework emphasizes smoothness and polynomial reproduction. Both sets of criteria have proven useful in applications, and interesting research synthesizing these perspectives is underway.

1.4 A BASIC SUBBAND IMAGE CODER

Three basic components underly current subband coders: a decorrelating transform, a quantization procedure, and entropy coding. This structure is a legacy of traditional transform coding, and has been with subband image coding from its earliest days [21, 22]. Before discussing state-of-the-art coders (and their advanced features) in the next sections, we will describe a basic subband coder and discuss issues in the design of its components.

1.4.1 Choice of Basis

Deciding on the optimal basis to use for image coding is a difficult problem. A number of design criteria, including smoothness, accuracy of approximation, size of support, and filter frequency selectivity are known to be important. However, the best combination of these features is not known.

The simplest form of basis for images is a separable basis formed from products of one dimensional filters. The problem of basis design is much simpler in one dimension, and almost all current coders employ separable transforms. Although the two-dimensional design problem is not as well understood, recent work of Sweldens and Kovačević [23] simplifies the design of non-separable bases, and such bases may prove more efficient than separable transforms.

Unser [24] shows that spline wavelets are attractive for coding applications based on approximation theoretic considerations. Experiments by Rioul [25] for orthogonal bases indicate that smoothness is an important consideration for compression. Experiments by Antonini *et al.* [26] find that both vanishing moments and smoothness are important, and for the filters tested they found that smoothness appeared to be slightly more important than the number of vanishing moments. Nonetheless, Vetterli and Herley [27] state that "the importance of regularity for signal processing applications is still an open question." The bases most commonly used in practice have between one and two continuous derivatives. Additional smoothness does not appear to yield significant improvements in coding results.

Villasenor *et al.* [28] have examined all minimum order biorthogonal filter banks with lengths ≤ 36. In addition to the criteria already mentioned, [28] also examines measures of oscillatory behavior and of the sensitivity of the coarse-scale approximations to the translations of the signal. The best filter found in these experiments was a 7/9-tap spline variant with less dissimilar lengths from [26], and this filter is one of the most commonly used in wavelet coders.

There is one caveat with regard to the results of the filter evaluation in [28]. Villasenor *et al.* compare peak signal to noise ratios generated by a simple transform coding scheme. The bit allocation scheme they use works well for orthogonal bases, but it can be improved upon considerably in the biorthogonal case. This inefficient bit allocation causes some promising biorthogonal filter sets to be overlooked.

For biorthogonal transforms, the squared error in the transform domain is not the same as the squared error in the original image. As a result, the problem of minimizing image error is considerably more difficult than in the orthogonal case. We can reduce image-domain errors by performing bit allocation using a weighted transform-domain error measure that we discuss in section 1.4.5. A number of other filters yield performance comparable to that of the 7/9 filter of [26] provided that we do bit allocation with a weighted error measure. One such basis is the Deslauriers-Dubuc interpolating wavelet of order 4 [29, 30], which has the advantage of having filter taps that are dyadic rationals. Others

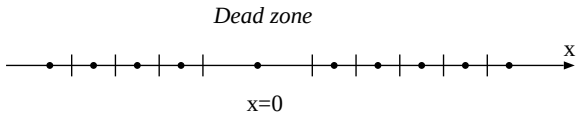

Figure 1.10 Dead-zone quantizer, with larger encoder partition around $x = 0$ (dead zone) and uniform quantization elsewhere.

are the 10/18 filters in [31], and the 28/28 filters designed with the software in [32].

One promising new set of filters has been developed by Balasingham and Ramstad [33]. Their design procedure combines classical filter design techniques with ideas from wavelet constructions and yields filters that perform better than the popular 7/9 filter set from [26].

1.4.2 Boundaries

Careful handling of image boundaries when performing the transform is essential for effective compression algorithms. Naive techniques for artificially extending images beyond given boundaries such as periodization or zero-padding lead to significant coding inefficiencies. For symmetrical bases, an effective strategy for handling boundaries is to extend the image via reflection [34]. Such an extension preserves continuity at the boundaries and usually leads to much smaller transform coefficients than if discontinuities were present at the boundaries. Brislawn [35] describes in detail procedures for non-expansive symmetric extensions of boundaries. An alternative approach is to modify the filter near the boundary. Boundary filters [36, 37] can be constructed that preserve filter orthogonality at boundaries. The lifting scheme [38] provides a related method for handling filtering near the boundaries.

1.4.3 Quantization

Most current subband coders employ scalar quantization for coding. There are two basic strategies for performing the scalar quantization stage. If we knew the distribution of coefficients for each subband in advance, the optimal strategy would be to use entropy-constrained Lloyd-Max quantizers for each subband. In general we do not have such knowledge, but we can provide a parametric description of coefficient distributions by sending side information. Coefficients in the high pass subbands of the transform are known *a priori* to be distributed as generalized Gaussians [39] centered around zero.

A much simpler quantizer that is commonly employed in practice is a uniform quantizer with a dead zone. The quantization bins, as shown in Figure 1.10, are of the form $[n\Delta, (n+1)\Delta)$ for $n \in \mathcal{Z}$ except for the central bin $[-\Delta, \Delta)$. Each bin is decoded to the value at its center in the simplest case, or to the centroid of the bin. In the case of asymptotically high rates, uniform quantization is optimal [40]. Although in practical regimes these dead-zone quantizers

are suboptimal, they work almost as well as Lloyd-Max coders when we decode to the bin centroids [41]. Moreover, dead-zone quantizers have the advantage that of being very low complexity and robust to changes in the distribution of coefficients in source. An additional advantage of these dead-zone quantizers is that they can be nested to produce an embedded bitstream following a procedure in [42].

1.4.4 Entropy Coding

Arithmetic coding provides a near-optimal entropy coding for the quantized coefficient values. The coder requires an estimate of the distribution of quantized coefficients. This estimate can be approximately specified by providing parameters for a generalized Gaussian or a Laplacian density. Alternatively the probabilities can be estimated online. Online adaptive estimation has the advantage of allowing coders to exploit local changes in image statistics. Efficient adaptive estimation procedures (context modeling) are discussed in [43, 44, 45, 46].

Because images are not jointly Gaussian random processes, the transform coefficients, although decorrelated, still contain considerable structure. The entropy coder can take advantage of some of this structure by conditioning the encodings on previously encoded values. Efficient context based modeling and entropy coding of wavelet coefficients can significantly improve the coding performance. In fact, several very competitive wavelet image coders are based on such techniques [42, 46, 47, 48].

1.4.5 Bit Allocation

The final question we need to address is that of how finely to quantize each subband. The general idea is to determine the number of bits b_j to devote to coding each subband j so that the total distortion $\sum_j D_j(b_j)$ is minimized subject to the constraint that $\sum_j b_j \leq B$. Here $D_j(b_j)$ is the amount of distortion incurred in coding subband j with b_j bits. When the functions $D_j(b)$ are known in closed form we can solve the problem using the Kuhn-Tucker conditions. One common practice is to approximate the functions $D_j(b)$ with the rate-distortion function for a Gaussian random variable. However, this approximation is not accurate at low bit rates. Better results may be obtained by measuring $D_j(b)$ for a range of values of b and then solving the constrained minimization problem using integer programming techniques. An algorithm of Shoham and Gersho [49] solves precisely this problem.

For biorthogonal wavelets we have the additional problem that squared error in the transform domain is not equal to squared error in the inverted image. Moulin [50] has formulated a multi-scale relaxation algorithm which provides an approximate solution to the allocation problem for this case. Moulin's algorithm yields substantially better results than the naive approach of minimizing squared error in the transform domain.

A simpler approach is to approximate the squared error in the image by weighting the squared errors in each subband. The weight w_j for subband j is

obtained as follows: we set a single coefficient in subband j to 1 and set all other wavelet coefficients to zero. We then invert this transform. The weight w_j is equal to the sum of the squares of the values in the resulting inverse transform. We allocate bits by minimizing the *weighted* sum $\sum_j w_j D_j(b_j)$ rather than the sum $\sum_j D_j(b_j)$. Further details may be found in Naveen and Woods [51]. This weighting procedure results in substantial coding improvements when using wavelets that are not very close to being orthogonal, such as the Deslauriers-Dubuc wavelets popularized by the lifting scheme [38]. The 7/9 tap filter set of [26], on the other hand, has weights that are all nearly 1, so this weighting provides little benefit.

1.4.6 Perceptually Weighted Error Measures

Our goal in lossy image coding is to minimize visual discrepancies between the original and compressed images. Measuring visual discrepancy is a difficult task. There has been a great deal of research on this problem, but because of the great complexity of the human visual system, no simple, accurate, and mathematically tractable measure has been found.

Our discussion up to this point has focused on minimizing squared error distortion in compressed images primarily because this error metric is mathematically convenient. The measure suffers from a number of deficits, however. For example, consider two images that are the same everywhere except in a small region. Even if the difference in this small region is large and highly visible, the mean squared error for the whole image will be small because the discrepancy is confined to a small region. Similarly, errors that are localized in straight lines, such as the blocking artifacts produced by the discrete cosine transform, are much more visually objectionable than squared error considerations alone indicate.

There is evidence that the human visual system makes use of a multiresolution image representation; see [52] for an overview. The eye is much more sensitive to errors in low frequencies than in high. As a result, we can improve the correspondence between our squared error metric and perceived error by weighting the errors in different subbands according to the eye's contrast sensitivity in a corresponding frequency range. Weights for the commonly used 7/9-tap filter set of [26] have been computed by Watson *et al.* in [53].

1.5 EXTENDING THE TRANSFORM CODER PARADIGM

The basic subband coder discussed in Section 1.4 is based on the traditional transform coding paradigm, namely decorrelation and scalar quantization of individual transform coefficients. The mathematical framework used in deriving the wavelet transform motivates compression algorithms that go beyond the traditional mechanisms used in transform coding. These important extensions are at the heart of modern coding algorithms of Sections 1.6 and 1.8. We take a moment here to discuss these extensions.

Conventional transform coding relies on energy compaction in an ordered set of transform coefficients, and quantizes those coefficients with a priority according to their order. This paradigm, while quite powerful, is based on several assumptions about images that are not always completely accurate. In particular, the Gaussian assumption breaks down for the joint distributions across image discontinuities. Mallat and Falzon [54] give the following example of how the Gaussian, high-rate analysis breaks down at low rates for non-Gaussian processes.

Let $Y[n]$ be a random N-vector defined by

$$Y[n] = \begin{cases} \text{X} & \text{if } n = P \\ \text{X} & \text{if } n = P + 1 (mod N) \\ 0 & \text{otherwise} \end{cases} \tag{1.8}$$

Here P is a random integer uniformly distributed between 0 and $N-1$ and X is a random variable that equals 1 or -1 each with probability $\frac{1}{2}$. X and P are independent. The vector Y has zero mean and a covariance matrix with entries

$$E\{Y[n]Y[m]\} = \begin{cases} \frac{2}{N} & \text{for } n = m \\ \frac{1}{N} & \text{for } |n-m| \in \{1, N-1\} \\ 0 & \text{otherwise} \end{cases} \tag{1.9}$$

The covariance matrix is circulant, so the KLT for this process is the simply the Fourier transform. The Fourier transform of Y is a very inefficient representation for coding Y. The energy at frequency k will be $|1 + e^{2\pi i \frac{k}{N}}|^2$ which means that the energy of Y is spread out over the entire low-frequency half of the Fourier basis with some spill-over into the high-frequency half. The KLT has "packed" the energy of the two non-zero coefficients of Y into roughly $\frac{N}{2}$ coefficients. It is obvious that Y was much more compact in its original form, and could be coded better without transformation: Only two coefficients in Y are non-zero, and we need only specify the values of these coefficients and their positions.

As suggested by the example above, the essence of the extensions to traditional transform coding is the idea of selection operators. Instead of quantizing the transform coefficients in a pre-determined order of priority, the wavelet framework lends itself to improvements, through judicious choice of which elements to code. This is made possible primarily because wavelet basis elements are spatially as well as spectrally compact. In parts of the image where the energy is spatially but not spectrally compact (like the example above) one can use selection operators to choose subsets of the transform coefficients that represent that signal efficiently. A most notable example is the Zerotree coder and its variants (Section 1.6).

More formally, the extension consists of dropping the constraint of linear image approximations, as the selection operator is nonlinear. The work of DeVore *et al.* [55] and of Mallat and Falzon [54] suggests that at low rates, the problem of image coding can be more effectively addressed as a problem in obtaining a *non-linear* image approximation. This idea leads to some important differences in coder implementation compared to the linear framework. For linear

Table 1.1 Peak signal to noise ratios in decibels for various coders

	Lena (b/p)			Barbara (b/p)		
Type of Coder	1.0	0.5	0.25	1.0	0.5	0.25
JPEG [56]	37.9	34.9	31.6	33.1	28.3	25.2
Optimized JPEG [57]	39.6	35.9	32.3	35.9	30.6	26.7
Baseline Wavelet [58]	39.4	36.2	33.2	34.6	29.5	26.6
Zerotree (Shapiro) [59]	39.6	36.3	33.2	35.1	30.5	26.8
Zerotree (Said-Pearlman) [60]	40.5	37.2	34.1	36.9	31.7	27.8
Zerotree (R-D optimized) [61]	40.5	37.4	34.3	37.0	31.3	27.2
Frequency-adaptive [62]	39.3	36.4	33.4	36.4	31.8	28.2
Space-frequency adaptive [63]	40.1	36.9	33.8	37.0	32.3	28.7
Frequency-adaptive + Zerotrees [64]	40.6	37.4	34.4	37.7	33.1	29.3
TCQ subband [65]	41.1	37.7	34.3	–	–	–
TCQ + zerotrees [66]	41.2	37.9	34.8	–	–	–
Bkwd. mixture estimation [67]	41.0	37.7	34.6	–	–	–
Context modeling (Chrysafis-Ortega) [48]	40.9	37.7	34.6	–	–	–
Context modeling (Wu) [46]	40.8	37.7	34.6	–	–	–

approximations, Theorems 1 and 2 in Section 1.3.1 suggest that at low rates we should approximate our images using a fixed subset of the Karhunen-Loève basis vectors. We set a fixed set of transform coefficients to zero, namely the coefficients corresponding to the smallest eigenvalues of the covariance matrix. The non-linear approximation idea, on the other hand, is to approximate images using a subset of basis functions that are selected adaptively based on the given image. Information describing the particular set of basis functions used for the approximation, called a significance map, is sent as side information. In Section 1.6 we describe zerotrees, a very important data structure used to efficiently encode significance maps.

Our example suggests that a second important assumption to relax is that our images come from a single jointly Gaussian source. We can obtain better energy compaction by optimizing our transform to the particular image at hand rather than to the global ensemble of images. Frequency-adaptive and space/frequency-adaptive coders decompose images over a large library of different bases and choose an energy-packing transform that is adapted to the image itself. We describe these adaptive coders in Section 1.7.

The selection operator that characterizes the extension to the transform coder paradigm generates information that needs to be conveyed to the decoder as "side information". This side information can be in the form of zerotrees, or more generally energy classes. Backward mixture estimation represents a different approach: it assumes that the side information is largely redundant and can be estimated from the causal data. By cutting down on the transmitted side information, these algorithms achieve a remarkable degree of performance and efficiency.

Figure 1.11 Compression of the 512×512 Barbara test image at 0.25 bits per pixel. Top left: original image. Top right: baseline JPEG, PSNR = 24.4 dB. Bottom left: baseline wavelet transform coder [58], PSNR = 26.6 dB. Bottom right: Said and Pearlman zerotree coder, PSNR = 27.6 dB.

For reference, Table 1.1 provides a comparison of the peak signal to noise ratios for the coders we discuss. The test images are the 512×512 Lena image and the 512×512 Barbara image. Figure 1.11 shows the Barbara image as compressed by JPEG, a baseline wavelet transform coder, and the zerotree coder of Said and Pearlman [60]. The Barbara image is particularly difficult to code, and we have compressed the image at a low rate to emphasize coder errors. The blocking artifacts produced by the discrete cosine transform are highly visible in the image on the top right. The difference between the two wavelet coded images is more subtle but quite visible at close range. Because of the more efficient coefficient encoding (to be discussed below), the zerotree-coded image has much sharper edges and better preserves the striped texture than does the baseline transform coder.

1.6 ZEROTREE CODING

The rate-distortion analysis of the previous sections showed that optimal bitrate allocation is achieved when the signal is divided into subbands such that each subband contains a "white" signal. It was also shown that for typical signals of interest, this leads to narrower bands in the low frequencies and wider bands in the high frequencies. Hence, wavelet transforms have very good energy compaction properties.

This energy compaction leads to efficient utilization of scalar quantizers. However, a cursory examination of the transform in Figure 1.12 shows that a significant amount of structure is present, particularly in the fine scale coefficients. Wherever there is structure, there is room for compression, and advanced wavelet compression algorithms all address this structure in the higher frequency subbands.

One of the most prevalent approaches to this problem is based on exploiting the relationships of the wavelet coefficients across bands. A direct visual inspection indicates that large areas in the high frequency bands have little or no energy, and the small areas that have significant energy are similar in shape and location, across different bands. These high-energy areas stem from poor energy compaction close to the edges of the original image. Flat and slowly varying regions in the original image are well-described by the low-frequency basis elements of the wavelet transform (hence leading to high energy compaction). At the edge locations, however, low-frequency basis elements cannot describe the signal adequately, and some of the energy leaks into high-frequency coefficients. This happens similarly at all scales, thus the high-energy high-frequency coefficients representing the edges in the image have the same shape.

Our *a priori* knowledge that images of interest are formed mainly from flat areas, textures, and edges, allows us to take advantage of the resulting cross-band structure. Zerotree coders combine the idea of cross-band correlation with the notion of coding zeros jointly (which we saw previously in the case of JPEG), to generate very powerful compression algorithms.

The first instance of the implementation of zerotrees is due to Lewis and Knowles [68]. In their algorithm the image is represented by a tree-structured

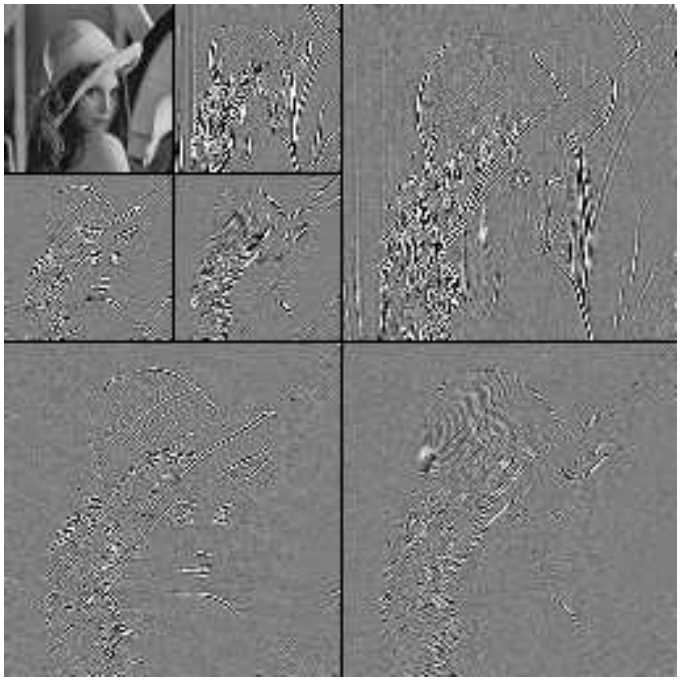

Figure 1.12 Wavelet transform of the image "Lena."

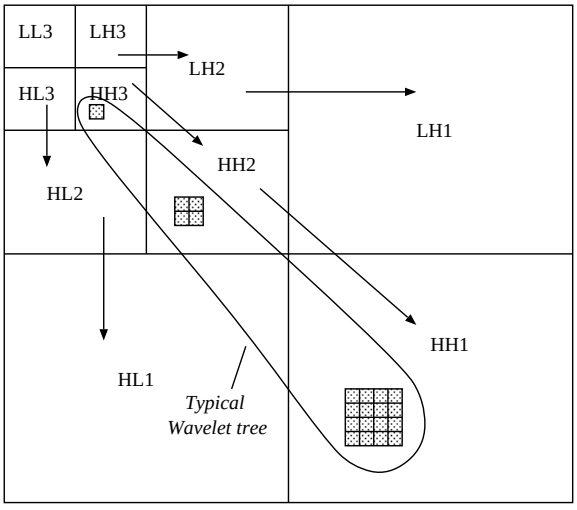

Figure 1.13 Space-frequency structure of wavelet transform

data construct (Figure 1.13). This data structure is implied by a dyadic discrete wavelet transform (Figure 1.9) in two dimensions. The root node of the tree represents the coefficient at the lowest frequency, which is the parent of three nodes. Nodes inside the tree correspond to wavelet coefficients at a frequency band determined by their height in the tree. Each of these coefficients has four children, which correspond to the wavelets at the next finer scale having the same location in space. These four coefficients represent the four phases of the higher resolution basis elements at that location. At the bottom of the data structure lie the leaf nodes, which have no children.

Note that there exist three such quadtrees for each coefficient in the low frequency band. Each of these three trees corresponds to one of three filtering orderings: there is one tree consisting entirely of coefficients arising from horizontal high-pass, vertical low-pass operation (HL); one for horizontal low-pass, vertical high-pass (LH), and one for high-pass in both directions (HH).

The zerotree quantization model used by Lewis and Knowles was arrived at by observing that often when a wavelet coefficient is small, its children on the wavelet tree are also small. This phenomenon happens because significant coefficients arise from edges and texture, which are local. It is not difficult to see that this is a form of conditioning. Lewis and Knowles took this conditioning to the limit, and assumed that insignificant parent nodes always imply insignificant child nodes. A tree or subtree that contains (or is assumed to contain) only insignificant coefficients is known as a zerotree.

The Lewis and Knowles coder achieves its compression ratios by joint coding of zeros. For efficient run-length coding, one needs to first find a conducive data structure, e.g. the zig-zag scan in JPEG. Perhaps the most significant contribution of this work was to realize that wavelet domain data provide an excellent context for run-length coding: not only are large run lengths of zeros generated, but also there is no need to transmit the length of zero runs, because they are assumed to automatically terminate at the leaf nodes of the tree. Much the same as in JPEG, this is a form of joint vector/scalar quantization. Each individual (significant) coefficient is quantized separately, but the symbols corresponding to small coefficients in fact are representing a vector consisting of that element and the zero run that follows it to the bottom of the tree.

1.6.1 The Shapiro and Said-Pearlman Coders

The Lewis and Knowles algorithm, while capturing the basic ideas inherent in many of the later coders, was incomplete. It had all the intuition that lies at the heart of more advanced zerotree coders, but did not efficiently specify significance maps, which is crucial to the performance of wavelet coders.

A significance map is a binary function whose value determines whether each coefficient is significant or not. If not significant, a coefficient is assumed to quantize to zero. Hence a decoder that knows the significance map needs no further information about that coefficient. Otherwise, the coefficient is quantized to a non-zero value. The method of Lewis and Knowles does not generate a significance map from the actual data, but uses one implicitly, based

on *a priori* assumptions on the structure of the data, namely that insignificant parent nodes imply insignificant child nodes. On the infrequent occasions when this assumption does not hold, a high price is paid in terms of distortion. The methods to be discussed below make use of the fact that, by using a small number of bits to correct mistakes in our assumptions about the occurrences of zerotrees, we can reduce the coded image distortion considerably.

The first algorithm of this family is due to Shapiro [59] and is known as the embedded zerotree wavelet (EZW) algorithm. Shapiro's coder was based on transmitting both the non-zero data and a significance map. The bits needed to specify a significance map can easily dominate the coder output, especially at lower bitrates. However, there is a great deal of redundancy in a general significance map for visual data, and the bitrates for its representation can be kept in check by conditioning the map values at each node of the tree on the corresponding value at the parent node. Whenever an insignificant parent node is observed, it is highly likely that the descendents are also insignificant. Therefore, most of the time, a "zerotree" significance map symbol is generated. But because p, the probability of this event, is close to 1, its information content, $-p \log p$, is very small. So most of the time, a very small amount of information is transmitted, and this keeps the average bitrate needed for the significance map relatively small.

Once in a while, one or more of the children of an insignificant node will be significant. In that case, a symbol for "isolated zero" is transmitted. The likelihood of this event is lower, and thus the bitrate for conveying this information is higher. But it is essential to pay this price to avoid losing significant information down the tree and therefore generating large distortions.

In summary, the Shapiro algorithm uses three symbols for significance maps: zerotree, isolated zero, or significant value. But using this structure, and by conditionally entropy coding these symbols, the coder achieves very good rate-distortion performance.

In addition, Shapiro's coder also generates an embedded code. Coders that generate embedded codes are said to have the *progressive transmission* or *successive refinement* property. Successive refinement consists of first approximating the image with a few bits of data, and then improving the approximation as more and more information is supplied. An embedded code has the property that for two given rates $R_1 > R_2$, the rate-R_2 code is a prefix to the rate-R_1 code. Such codes are of great practical interest for the following reasons:

- The encoder can easily achieve a precise bitrate by continuing to output bits when it reaches the desired rate.
- The decoder can cease decoding at any given point, generating an image that is the best representation possible with the decoded number of bits. This is of practical interest for broadcast applications where multiple decoders with varying computational, display, and bandwidth capabilities attempt to receive the same bitstream. With an embedded code, each receiver can decode the passing bitstream according to its particular needs and capabilities.

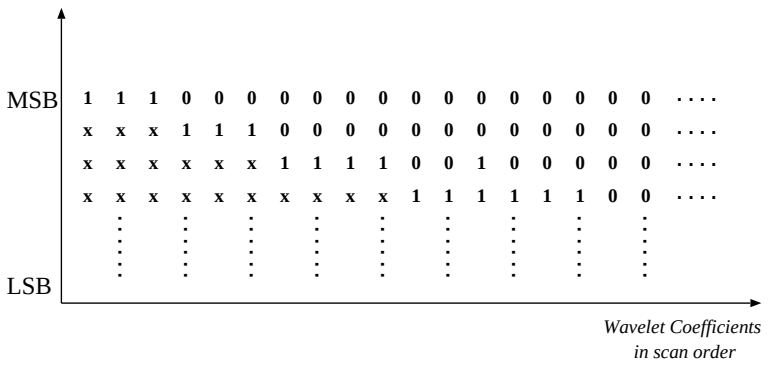

Figure 1.14 Bit plane profile for raster scan ordered wavelet coefficients

- Embedded codes are also very useful for indexing and browsing, where only a rough approximation is sufficient for deciding whether the image needs to be decoded or received in full. The process of screening images can be speeded up considerably by using embedded codes: after decoding only a small portion of the code, one knows if the target image is present. If not, decoding is aborted and the next image is requested, making it possible to screen a large number of images quickly. Once the desired image is located, the complete image is decoded.

Shapiro's method generates an embedded code by using a bit-slice approach (see Figure 1.14). First, the wavelet coefficients of the image are indexed into a one-dimensional array, according to their order of importance. This order places lower frequency bands before higher frequency bands since they have more energy, and coefficients within each band appear in a raster scan order. The bit-slice code is generated by scanning this one-dimensional array, comparing each coefficient with a threshold T. This initial scan provides the decoder with sufficient information to recover the most significant bit slice. In the next pass, our information about each coefficient is refined to a resolution of $T/2$, and the pass generates another bit slice of information. This process is repeated until there are no more slices to code.

Figure 1.14 shows that the upper bit slices contain a great many zeros because there are many coefficients below the threshold. The role of zerotree coding is to avoid transmitting all these zeros. Once a zerotree symbol is transmitted, we know that all the descendent coefficients are zero, so no information is transmitted for them. In effect, zerotrees are a clever form of run-length coding, where the coefficients are ordered in a way to generate longer run lengths (more efficient) as well as making the runs self-terminating, so the length of the runs need not be transmitted.

The zerotree symbols (with high probability and small code length) can be transmitted again and again for a given coefficient, until it rises above the sinking threshold, at which point it will be tagged as a significant coefficient.

After this point, no more zerotree information will be transmitted for this coefficient.

To achieve embeddedness, Shapiro uses a clever method of encoding the sign of the wavelet coefficients with the significance information. There are also further details of the priority of wavelet coefficients, the bit-slice coding, and adaptive arithmetic coding of quantized values (entropy coding), which we will not pursue further in this review. The interested reader is referred to [59] for more details.

Said and Pearlman [60] have produced an enhanced implementation of the zerotree algorithm, known as Set Partitioning in Hierarchical Trees (SPIHT). Their method is based on the same premises as the Shapiro algorithm, but with more attention to detail. The public domain version of this coder is very fast, and improves the performance of EZW by 0.3-0.6 dB. This gain is mostly due to the fact that the original zerotree algorithms allow special symbols only for single zerotrees, while in reality, there are other sets of zeros that appear with sufficient frequency to warrant special symbols of their own. In particular, the Said-Pearlman coder provides symbols for combinations of parallel zerotrees.

Davis and Chawla [69] have shown that both the Shapiro and the Said and Pearlman coders are members of a large family of tree-structured significance mapping schemes. They provide a theoretical framework that explains in more detail the performance of these coders and describe an algorithm for selecting a member of this family of significance maps that is optimized for a given image or class of images.

1.6.2 Zerotrees and Rate-Distortion Optimization

In the previous coders, zerotrees were used only when they were detected in the actual data. But consider for the moment the following hypothetical example: assume that in an image, there is a wide area of little activity, so that in the corresponding location of the wavelet coefficients there exists a large group of insignificant values. Ordinarily, this would warrant the use of a big zerotree and a low expenditure of bitrate over that area. Suppose, however, that there is a one-pixel discontinuity in the middle of the area, such that at the bottom of the would-be zerotree, there is one significant coefficient. The algorithms described so far would prohibit the use of a zerotree for the entire area.

Inaccurate representation of a single pixel will change the average distortion in the image only by a small amount. In our example we can gain significant coding efficiency by ignoring the single significant pixel so that we can use a large zerotree. We need a way to determine the circumstances under which we should ignore significant coefficients in this manner.

The specification of a zerotree for a group of wavelet coefficient is a form of quantization. Generally, the values of the pixels we code with zerotrees are non-zero, but in using a zerotree we specify that they be decoded as zeros. Non-zerotree wavelet coefficients (significant values) are also quantized, using scalar quantizers. If we saves bitrate by specifying larger zerotrees, as in the hypothetical example above, the rate that was saved can be assigned to the

scalar quantizers of the remaining coefficients, thus quantizing them more accurately. Therefore, we have a choice in allocating the bitrate among two types of quantization. The question is, if we are given a unit of rate to use in coding, where should it be invested so that the corresponding reduction in distortion is maximized?

This question, in the context of zerotree wavelet coding, was addressed by Xiong *et al.* [61], using well-known bit allocation techniques [8]. The central result for optimal bit allocation states that, in the optimal state, the slope of the operational rate-distortion curves of all quantizers are equal. This result is intuitive and easy to understand. The slope of the operational rate-distortion function for each quantizer tells us how many units of distortion we add/eliminate for each unit of rate we eliminate/add. If one of the quantizers has a smaller R-D slope, meaning that it is giving us less distortion reduction for our bits spent, we can take bits away from this quantizer (i.e. we can reduce its step size) and give them to the other, more efficient quantizers. We continue to do so until all quantizers have an equal slope.

Obviously, specification of zerotrees affects the quantization levels of non-zero coefficients because total available rate is limited. Conversely, specifying quantization levels will affect the choice of zerotrees because it affects the incremental distortion between zerotree quantization and scalar quantization. Therefore, an iterative algorithm is needed for rate-distortion optimization. In phase one, the uniform scalar quantizers are fixed, and optimal zerotrees are chosen. In phase two, zerotrees are fixed and the quantization level of uniform scalar quantizers is optimized. This algorithm is guaranteed to converge to a local optimum [61].

There are further details of this algorithm involving prediction and description of zerotrees, which we leave out of the current discussion. The advantage of this method is mainly in performance, compared to both EZW and SPIHT (the latter only slightly). The main disadvantages of this method are its complexity, and perhaps more importantly, that it does not generate an embedded bitstream.

1.7 FREQUENCY, SPACE-FREQUENCY ADAPTIVE CODERS

1.7.1 Wavelet Packets

The wavelet transform does a good job of decorrelating image pixels in practice, especially when images have power spectra that decay approximately uniformly and exponentially. However, for images with non-exponential rates of spectral decay and for images which have concentrated peaks in the spectra away from DC, we can do considerably better.

Our analysis of Section 1.3.2 suggests that the optimal subband decomposition for an image is one for which the spectrum in each subband is approximately flat. The octave-band decomposition produced by the wavelet transform produces nearly flat spectra for exponentially decaying spectra. The Barbara test image shown in Figure 1.11 contains a narrow-band component at high fre-

quencies that comes from the tablecloth and the striped clothing. Fingerprint images contain similar narrow-band high frequency components.

The best basis algorithm, developed by Coifman and Wickerhauser [70], provides an efficient way to find a fast, wavelet-like transform that provides a good energy compaction for a given image. The new basis functions are not wavelets but rather *wavelet packets* [71].

The basic idea of wavelet packets is best seen in the frequency domain. Each step of the wavelet transform splits the current low frequency subband into two subbands of equal width, one high-pass and one low-pass. With wavelet packets there is a new degree of freedom in the transform. Again there are N stages to the transform for a signal of length 2^N, but at each stage we have the option of splitting the low-pass subband, the high-pass subband, both, or neither. The high and low pass filters used in each case are the same filters used in the wavelet transform. In fact, the wavelet transform is the special case of a wavelet packet transform in which we always split the low-pass subband. With this increased flexibility we can generate 2^N possible different transforms in 1-D. The possible transforms give rise to all possible dyadic partitions of the frequency axis. The increased flexibility does not lead to a large increase in complexity; the worst-case complexity for a wavelet packet transform is $O(N \log N)$.

1.7.2 Frequency Adaptive Coders

The *best basis algorithm* is a fast algorithm for minimizing an additive cost function over the set of all wavelet packet bases. Our analysis of transform coding for Gaussian random processes suggests that we select the basis that maximizes the transform coding gain. The approximation theoretic arguments of Mallat and Falzon [54] suggest that at low bit rates the basis that maximizes the number of coefficients below a given threshold is the best choice. The best basis paradigm can accommodate both of these choices. See [72] for an excellent introduction to wavelet packets and the best basis algorithm. Ramchandran and Vetterli [62] describe an algorithm for finding the best wavelet packet basis for coding a given image using rate-distortion criteria.

An important application of this wavelet-packet transform optimization is the FBI Wavelet/Scalar Quantization Standard for fingerprint compression. The standard uses a wavelet packet decomposition for the transform stage of the encoder [73]. The transform used is fixed for all fingerprints, however, so the FBI coder is a first-generation linear coder.

The benefits of customizing the transform on a per-image basis depend considerably on the image. For the Lena test image the improvement in peak signal to noise ratio is modest, ranging from 0.1 dB at 1 bit per pixel to 0.25 dB at 0.25 bits per pixel. This is because the octave band partitions of the spectrum of the Lena image are nearly flat. The Barbara image (see Figure 1.11), on the other hand, has a narrow-band peak in the spectrum at high frequencies. Consequently, the PSNR increases by roughly 2 dB over the same range of bitrates [62]. Further impressive gains are obtained by combining the adaptive transform with a zerotree structure [64].

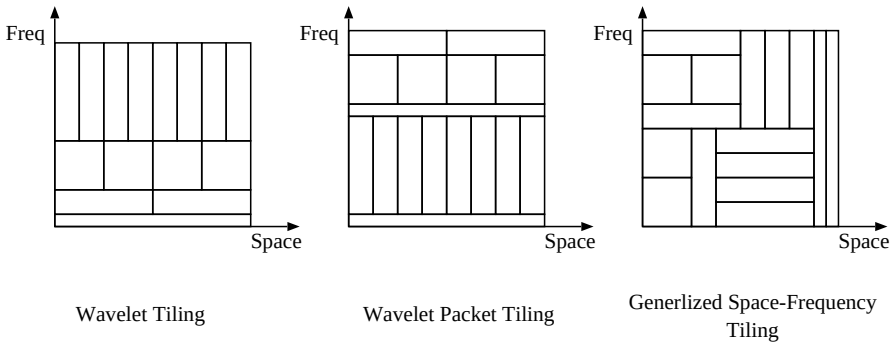

Figure 1.15 Wavelets, wavelet packets, and generalized time-frequency tiling

1.7.3 Space-Frequency Adaptive Coders

The best basis algorithm is not limited only to adaptive segmentation of the frequency domain. Related algorithms permit joint time and frequency segmentations. The simplest of these algorithms performs adapted frequency segmentations over regions of the image selected through a quadtree decomposition procedure [74, 75]. More complicated algorithms provide combinations of spatially varying frequency decompositions and frequency varying spatial decompositions [63]. These jointly adaptive algorithms work particularly well for highly nonstationary images.

The primary disadvantage of these spatially adaptive schemes are that the pre-computation requirements are much greater than for the frequency adaptive coders, and the search is also much larger. A second disadvantage is that both spatial and frequency adaptivity are limited to dyadic partitions. A limitation of this sort is necessary for keeping the complexity manageable, but dyadic partitions are not in general the best ones. Figure 1.15 shows an example of the time-frequency tiling of wavelets, wavelet packets, and space-frequency adaptive basis.

1.8 UTILIZING INTRA-BAND DEPENDENCIES

The development of the EZW coder motivated a flurry of activity in the area of zerotree wavelet algorithms. The inherent simplicity of the zerotree data structure, its computational advantages, as well as the potential for generating an embedded bitstream were all very attractive to the coding community. Zerotree algorithms were developed for a variety of applications, and many modifications and enhancements to the algorithm were devised, as described in Section 1.6.

With all the excitement incited by the discovery of EZW, it is easy to automatically assume that zerotree structures, or more generally inter-band dependencies, should be the focal point of efficient subband image compression algorithms. However, some of the best performing subband image coders known

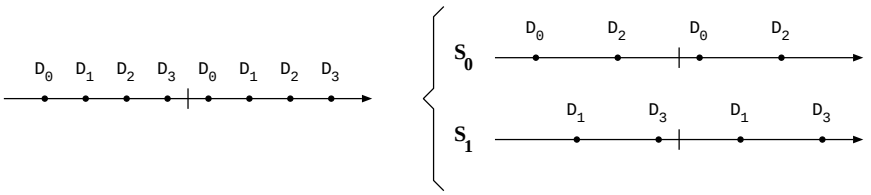

Figure 1.16 TCQ sets and supersets

today are not based on zerotrees. In this section, we explore two methods that utilize intra-band dependencies. One of them uses the concept of Trellis Coded Quantization (TCQ). The other uses both inter- and intra-band information, and is based on a recursive estimation of the variance of the wavelet coefficients. Both of them yield excellent coding results.

1.8.1 Trellis Coded Quantization

Trellis Coded Quantization (TCQ) [76] is a fast and effective method of quantizing random variables. Trellis coding exploits correlations between variables. More interestingly, it can use non-rectangular quantizer cells that give it quantization efficiencies not attainable by scalar quantizers. TCQ grew out of the ground-breaking work of Ungerboeck [77] in trellis coded modulation.

The basic idea behind TCQ is the following: Assume that we want to quantize a stationary, memoryless uniform source at the rate of R bits per sample. Performing quantization directly on this uniform source would require an optimum scalar quantizer with 2^N reproduction levels (symbols). The idea behind TCQ is to first quantize the source more finely, with 2^{R+k} symbols. Of course this would exceed the allocated rate, so we cannot have a free choice of symbols at all times.

In our example we take $k = 1$. The scalar codebook of 2^{R+1} symbols is partitioned into subsets of 2^{R-1} symbols each, generating four sets. In our example $R = 2$; see Figure 1.16. The subsets are designed such that each of them represents reproduction points of a coarser, rate-$(R-1)$ quantizer. The four subsets are designated D_0, D_1, D_2, and D_3. Also, define $S_0 = D_0 \bigcup D_2$ and $S_1 = D_1 \bigcup D_3$, where S_0 and S_1 are known as *supersets*.

Obviously, the rate constraint prohibits the specification of an arbitrary symbol out of 2^{R+1} symbols. However, it is possible to exactly specify, with R bits, one element out of either S_0 or S_1. At each sample, assuming we know which one of the supersets to use, one bit can be used to determine the active subset, and $R-1$ bits to specify a codeword from the subset. The choice of superset is determined by the state of a finite state machine, described by a suitable trellis. An example of such a trellis, with eight states, is given in Figure 1.17. The subsets $\{D_0, D_1, D_2, D_3\}$ are also used to label the branches of the trellis, so the same bit that specifies the subset (at a given state) also determines the next state of the trellis.

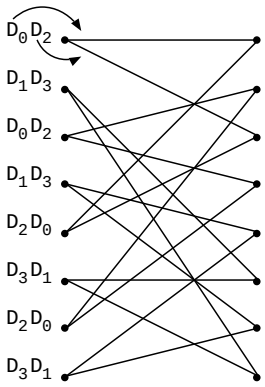

Figure 1.17 8-state TCQ trellis with subset labeling. The bits that specify the sets within the superset also dictate the path through the trellis.

Encoding is achieved by spending one bit per sample on specifying the path through the trellis, while the remaining $R-1$ bits specify a codeword out of the active subset. It may seem that we are back to a non-optimal rate-R quantizer (either S_0 or S_1). So why all this effort? The answer is that we have more codewords than a rate-R quantizer, because there is some freedom of choosing from symbols of either S_0 or S_1. Of course this choice is not completely free: the decision made at each sample is linked to decisions made at past and future sample points, through the permissible paths of the trellis. But it is this additional flexibility that leads to the improved performance. Availability of both S_0 and S_1 means that the reproduction levels of the quantizer are, in effect, allowed to "slide around" and fit themselves to the data, subject to the permissible paths on the trellis.

The standard version of TCQ is not particularly suitable for image coding, because its performance degrades quickly at low rates. This is due partially to the fact that one bit per sample is used to encode the trellis alone, while interesting rates for image coding are mostly below one bit per sample. Entropy constrained TCQ (ECTCQ) improves the performance of TCQ at low rates. In particular, a version of ECTCQ due to Marcellin [78] addresses two key issues: reducing the rate used to represent the trellis (the so-called "state entropy"), and ensuring that zero can be used as an output codeword with high probability. The codebooks are designed using the algorithm and encoding rule from [79].

1.8.2 TCQ Subband Coders

Consider a subband decomposition of an image, and assume that the subbands are well represented by a *non-stationary* random process X, whose samples X_i are taken from distributions with variances σ_i^2. One can compute an "average variance" over the entire random process and perform conventional optimal quantization. But better performance is possible by sending overhead informa-

tion about the variance of each sample, and quantizing it optimally according to its own p.d.f.

This basic idea was first proposed by Chen and Smith [80] for adaptive quantization of DCT coefficients. In their paper, Chen and Smith proposed to divide all DCT coefficients into four groups according to their "activity level", i.e. variance, and code each coefficient with an optimal quantizer designed for its group. The question of how to partition coefficients into groups was not addressed, however, and [80] arbitrarily chose to form groups with equal population.[4]

However, one can show that equally populated groups are not a always a good choice. Suppose that we want to classify the samples into J groups, and that all samples assigned to a given class $i \in \{1, ..., J\}$ are grouped into a source X_i. Let the total number of samples assigned to X_i be N_i, and the total number of samples in all groups be N. Define $p_i = N_i/N$ to be the probability of a sample belonging to the source X_i. Encoding the source X_i at rate R_i results in a mean squared error distortion of the form [81]

$$D_i(R_i) = \epsilon_i^2 \, \sigma_i^2 \, 2^{-2R_i} \tag{1.10}$$

where ϵ_i is a constant depending on the shape of the pdf. The rate allocation problem can now be solved using a Lagrange multiplier approach, much in the same way as was shown for optimal linear transforms, resulting in the following optimal rates:

$$R_i = \frac{R}{J} + \frac{1}{2} \log_2 \frac{\epsilon_i^2 \, \sigma_i^2}{\prod_{j=1}^{J} (\epsilon_j^2 \, \sigma_j^2)^{p_j}} \tag{1.11}$$

where R is the total rate and R_i are the rates assigned to each group. *Classification gain* is defined as the ratio of the quantization error of the original signal X, divided by that of the optimally bit-allocated classified version.

$$G_c = \frac{\epsilon^2 \, \sigma^2}{\prod_{j=1}^{J} (\epsilon_j^2 \, \sigma_j^2)^{p_j}} \tag{1.12}$$

One aims to maximize this gain over $\{p_i\}$. It is not unexpected that the optimization process can often yield non-uniform $\{p_i\}$, resulting in unequal population of the classification groups. It is noteworthy that non-uniform populations not only have better classification gain in general, but also lower overhead: Compared to a uniform $\{p_i\}$, any other distribution has smaller entropy, which implies smaller side information to specify the classes.

[4] If for a moment, we disregard the overhead information, the problem of partitioning the coefficients bears a strong resemblance to the problem of best linear transform. Both operations, namely the linear transform and partitioning, conserve energy. The goal in both is to minimize overall distortion through optimal allocation of a finite rate. Not surprisingly, the solution techniques are similar (Lagrange multipliers), and they both generate sets with maximum separation between low and high energies (maximum arithmetic to geometric mean ratio).

The classification gain is defined for X_i taken from one subband. A generalization of this result in [65] combines it with the conventional *coding gain* of the subbands. Another refinement takes into account the side information required for classification. The coding algorithm then optimizes the resulting expression to determine the classifications. ECTCQ is then used for final coding.

Practical implementation of this algorithm requires attention to a great many details, for which the interested reader is referred to [65]. For example, the classification maps determine energy levels of the signal, which are related to the location of the edges in the image, and are thus related in different subbands. A variety of methods can be used to reduce the overhead information (in fact, the coder to be discussed in the next section makes the management of side information the focus of its efforts) Other issues include alternative measures for classification, and the usage of arithmetic coded TCQ. The coding results of the ECTCQ based subband coding are some of the best currently available in the literature, although the computational complexity of these algorithms is also considerably greater than the other methods presented in this paper.

Yet better performance is possible (at the expense of higher complexity) by combining ideas from space-frequency quantization (SFQ) and trellis coding. Trellis coded space-frequency quantization (TCSFQ) [66] is the result of this combination. The basic idea of TCSFQ is to throw away a subset of wavelet coefficients and apply TCQ to the rest. TCSFQ can thus be thought of as taking the "best-of-the-best" from both SFQ and TCQ.

The SFQ algorithm in [61] takes advantage of the space-frequency characteristics of wavelet coefficients. It prunes a subset of spatial tree coefficients (i.e. setting the coefficients to zero) and uses scalar quantization on the rest. Optimal pruning in SFQ is achieved in a rate-distortion sense. SFQ thus uses an explicit form of subband classification, which has been shown to provide significant gain in wavelet image coding. Subband classification provides context models for both quantization and entropy coding.

SFQ only realizes the classification gain [65] with a single uniform scalar quantization applied on the non-pruned subset. Using TCQ on this set will further exploit the packing gain [82] of the trellis code, thus improving the coding performance. When combined with the conditional entropy coding scheme in [46], TCSFQ offers very good coding performance (see Table 1.1).

1.8.3 Context and Mixture Modeling

A common thread in successful subband and wavelet image coders is modeling of image subbands as random variables drawn from a mixture of distributions. For each sample, one needs to detect which p.d.f. of the mixture it is drawn from, and then quantize it according to that pdf. Since the decoder needs to know which element of the mixture was used for encoding, many algorithms send side information to the decoder. This side information becomes significant, especially at low bitrates, so that efficient management of it is pivotal to the success of the image coder.

All subband and wavelet coding algorithms discussed so far use this idea in one way or another. They only differ in the constraints they put on side information so that it can be coded efficiently. For example, zerotrees are a clever way of indicating side information. The data is assumed from a mixture of very low energy (zero set) and high energy random variables, and the zero sets are assumed to have a tree structure.

The TCQ subband coders discussed in the last section also use the same idea. Different classes represent different energies in the subbands, and are transmitted as overhead. In [65], several methods are discussed to compress the side information, again based on geometrical constraints on the constituent elements of the mixture (energy classes).

A completely different approach to the problem of handling information overhead is explored in quantization via mixture modeling [67]. The version developed in [67] is named Estimation Quantization (EQ) by the authors, and is the one that we present in the following. We will refer to the the aggregate class as *backward mixture-estimation encoding* (BMEE).

BMEE models the wavelet subband coefficients as non-stationary generalized Gaussian, whose non-stationarity is manifested by a slowly varying variance (energy) in each band. Because the energy varies slowly, it can be predicted from causal neighboring coefficients. Therefore, unlike previous methods, BMEE does not send the bulk of mixture information as overhead, but attempts to recover it at the decoder from already transmitted data, hence the designation "backward". BMEE assumes that the causal neighborhood of a subband coefficient (including parents in a subband tree) has the same energy (variance) as the coefficient itself. The estimate of energy is found by applying a maximum likelihood method to a training set formed by the causal neighborhood.

Similar to other recursive algorithms that involve quantization, BMEE has to contend with the problem of stability and drift. Specifically, the decoder has access only to quantized coefficients, therefore the estimator of energy at the encoder can only use quantized coefficients. Otherwise, the estimates at the encoder and decoder will vary, resulting in drift problems. This presents the added difficulty of estimating variances from *quantized* causal coefficients. BMEE incorporates the quantization of the coefficients into the maximum likelihood estimation of the variance.

The quantization itself is performed with a dead-zone uniform quantizer (see Figure 1.10). This quantizer offers a good approximation to entropy constrained quantization of generalized Gaussian signals. The dead-zone and step sizes of the quantizers are determined through a Lagrange multiplier optimization technique, which was introduced in the section on optimal rate allocation. This optimization is performed off-line, once each for a variety of encoding rates and shape parameters, and the results are stored in a look-up table. This approach is to be credited for the speed of the algorithm, because no optimization need take place at the time of encoding the image.

Finally, the backward nature of the algorithm, combined with quantization, presents another challenge. All the elements in the causal neighborhood may sometimes quantize to zero. In that case, the current coefficient will also quantize to zero. This degenerate condition will propagate through the subband, making all coefficients on the causal side of this degeneracy equal to zero. To avoid this condition, BMEE provides for a mechanism to send side information to the receiver, whenever all neighboring elements are zero. This is accomplished by a preliminary pass through the coefficients, where the algorithm tries to "guess" which one of the coefficients will have degenerate neighborhoods, and assembles them to a set. From this set, a generalized Gaussian variance and shape parameter is computed and transmitted to the decoder. Every time a degenerate case happens, the encoder and decoder act based on this extra set of parameters, instead of using the backward estimation mode.

The BMEE coder is very fast, and especially in the low bitrate mode (less than 0.25 bits per pixel) is extremely competitive. This is likely to motivate a revisitation of the role of side information and the mechanism of its transmission in wavelet coders.

Once the quantization process is completed, another category of modeling is used for entropy coding. Context modeling in entropy coding [46, 48] attempts to estimate the probability distribution of the next symbol based on past samples. In this area, Wu's work [46] on conditional entropy coding of wavelets (CECOW) is noteworthy. CECOW utilizes a sophisticated modeling structure and seeks improvements in two directions: First, a straight forward increase in the order of models, compared to methods such as EZW and SPIHT. To avoid the problem of context dilution, CECOW determines the number of model parameters by adaptive context formation and minimum entropy quantization of contexts. Secondly, CECOW allows the shape and size of the context to vary among subbands, thus allowing more flexibility.

1.9 DISCUSSION AND SUMMARY

1.9.1 Discussion

Current research in image coding is progressing along a number of fronts on transform, quantization, and entropy coding.

At the most basic level, a new interpretation of the wavelet transform has appeared in the literature. This new theoretical framework, called the lifting scheme [30], provides a simpler and more flexible method for designing wavelets than standard Fourier-based methods. New families of non-separable wavelets constructed using lifting have the potential to improve coders. One very intriguing avenue for future research is the exploration of the nonlinear analogs of the wavelet transform that lifting makes possible. In particular, integer transforms are more easily designed with the lifting techniques, leading to efficient lossless compression, as well as computationally efficient lossy coders.

Recent developments of high performance subband/wavelet image coders (reviewed in this chapter) suggest that further improvements in performance

may be possible through a better understanding of the statistical properties of subband coefficients. Subband classification is an explicit way of modeling these coefficients in quantization, while context modeling in entropy coding is aimed at the same goal. If subband classification and context modeling can be jointly optimized, improvements may be achieved over the current state-of-the-art.

With the increased performance provided by the subband coders, efforts of the community are partially channeled to other issues in image coding, such as spatial scalability, lossy to lossless coding, region-of-interest coding, and error resilience. Error resilience image coding via joint source-channel coding is a very promising research direction. See for example [83], [84], [85] and [86].

The adoption of wavelet based coding to video signals presents special challenges. One can apply 2-D wavelet coding in combination to temporal prediction (motion estimated prediction), which will be a direct counterpart of current DCT-based video coding methods. It is also possible to consider the video signal as a three-dimensional array of data and attempt to compress it with 3-D wavelet analysis. 3-D wavelet video coding has been explored by a number of researchers (see the collection of papers in this area in [87]). This 3-D wavelet based approach presents difficulties that arise from the fundamental properties of the discrete wavelet transform. The discrete wavelet transform (as well as any subband decomposition) is a space-varying operator, due to the presence of decimation and interpolation. This space variance is not conducive to compact representation of video signals, as described below.

Video signals are best modeled by 2-D projections whose position in consecutive frames of the video signal varies by unknown amounts. Because vast amounts of information are repeated in this way, one can achieve considerable gain by representing the repeated information only once. This is the basis of motion compensated coding. However, since the wavelet representation of the same 2-D signal will vary once it is shifted[5], this redundancy is difficult to reproduce in the wavelet domain. A frequency domain study of the difficulties of 3-D wavelet coding of video is presented in [88], and leads to the same insights. Some attempts have also been made on applying 3-D wavelet coding on the residual 3-D data after motion compensation, but have met with indifferent success.

1.9.2 Summary

Image compression is governed by the general laws of information theory and specifically rate-distortion theory. However, these general laws are nonconstructive and more specific techniques of quantization theory are needed for the actual development of compression algorithms.

Vector quantization can theoretically attain the maximum achievable coding efficiency. However, VQ has three main impediments: computational complex-

[5]Unless the shift is exactly by a correct multiple of M samples, where M is the downsampling rate

ity, delay, and the curse of dimensionality. Transform coding techniques, in conjunction with entropy coding, capture important gains of VQ, while avoiding most of its difficulties.

Theoretically, the Karhunen-Loéve transform (KLT) is optimal for Gaussian processes, among block transforms. Approximations to the KLT, such as the DCT, have led to very successful image coding algorithms such as JPEG. However, even if one argues that image pixels can be individually Gaussian, they cannot be assumed to be jointly Gaussian, at least not across the image discontinuities. Image discontinuities are the place where traditional coders spend the most rate, and suffer the most distortion. This happens because traditional Fourier-type transforms (e.g., DCT) disperse the energy of discontinuous signals across many coefficients, while the compaction of energy in the transform domain is essential for good coding performance.

Smooth subband bases of compact support, in particular the wavelet transform, provide an elegant framework for signal representation in which both smooth areas and discontinuities can be represented compactly in the transform domain.

State of the art wavelet coders assume that image data comes from a source with fluctuating variance. Each of these coders provides a mechanism to express the local variance of the wavelet coefficients, and quantizes the coefficients optimally or near-optimally according to that variance. The individual wavelet coders vary in the way they estimate and transmit this variances to the decoder, as well as the strategies for quantizing according to that variance.

Zerotree coders assume a two-state structure for the variances: either negligible (zero) or otherwise. They send side information to the decoder to indicate the positions of the non-zero coefficients. This process yields a non-linear image approximation rather than the linear truncated KLT-based approximation motivated by our Gaussian model. The set of zero coefficients are expressed in terms of wavelet trees (Lewis & Knowles, Shapiro, others) or combinations thereof (Said & Pearlman). The zero sets are transmitted to the receiver as overhead, as well as the rest of the quantized data. Zerotree coders rely strongly on the dependency of data across scales of the wavelet transform.

Frequency-adaptive coders improve upon basic wavelet coders by adapting transforms according to the local inter-pixel correlation structure within an image. Local fluctuations in the correlation structure and in the variance can be addressed by spatially adapting the transform and by augmenting the optimized transforms with a zerotree structure.

Other wavelet coders use dependency of data within the bands (and sometimes across the bands as well). Coders based on Trellis Coded Quantization (TCQ) partition coefficients into a number of groups, according to their energy. For each coefficient, they estimate and/or transmit the group information as well as coding the value of the coefficient with TCQ, according to the nominal variance of the group. Another newly developed class of coders transmit only minimal variance information while achieving impressive coding results, indi-

cating that perhaps the variance information is more redundant than previously thought.

Subband transforms and the ideas arising from wavelet analysis have had an indelible effect on the theory and practice of image compression, and are likely to continue their dominant presence in image coding research in the near future.

References

[1] M. Vetterli and J. Kovačević, *Wavelets and Subband Coding*. Englewood Cliffs, NJ: Prentice Hall, 1995.

[2] P. Vaidyanathan, *Multirate systems and filter banks*. Englewood Cliffs, NJ: Prentice Hall, 1993.

[3] C. E. Shannon, "Coding theorems for a discrete source with a fidelity criterion," in *IRE Nat. Conv. Rec.*, vol. 4, pp. 142–163, Mar. 1959.

[4] S. Lloyd, "Least squares quantization in PCM." Unpublished Bell Labs Technical Note, Sept. 1957.

[5] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. IT-28, pp. 127–135, Mar. 1982.

[6] J. Max, "Quantizaing for minimum distortion," *IEEE Transactions on Information Theory*, pp. 7–12, Mar. 1960.

[7] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *IEEE Transactions on Communications*, vol. COM-26, pp. 84–95, Jan. 1980.

[8] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Boston, MA: Kluwer Academic Publishers, 1992.

[9] A. Gersho, "Asymptotically optimal block quantization," *IEEE Transactions on Information Theory*, vol. IT-25, pp. 373–380, July 1979.

[10] P. L. Zador, "Asymptotic quantization error of continuous signals and the quantization dimension," *IEEE Transactions on Information Theory*, vol. IT-28, pp. 139–149, Mar. 1982.

[11] G. Davis and A. Nosratinia, "Wavelet-based image coding: an overview," *Applid and Computational Control, Signals, and Circuits*, vol. 1, pp. 205–269, 1998.

[12] M. Unser, "An extension of the Karhunen-Loève transform for wavelets and perfect reconstruction filterbanks," in *SPIE Mathematical Imaging*, vol. 2034, pp. 45–56, 1993.

[13] R. M. Gray, "On the asymptotic eigenvalue distribution of Toeplitz matrices," *IEEE Transactions on Information Theory*, vol. IT-18, pp. 725–230, Nov. 1972.

[14] T. Berger, *Rate Distortion Theory.* Englewood Cliffs, NJ: Prentice Hall, 1971.

[15] A. Croisier, D. Esteban, and C. Galand, "Perfect channel splitting by use of interpolation/decimation/tree decomposition techniques," in *Proc. Int. Symp. on Information, Circuits and Systems*, (Patras, Greece), 1976.

[16] P. Vaidyanathan, "Theory and design of M-channel maximally decimated quadrature mirror filters with arbitrary M, having perfect reconstruction property," *IEEE Transactions on Acoust. Speech Signal Process.*, vol. ASSP-35, pp. 476–492, Apr. 1987.

[17] M. J. T. Smith and T. P. Barnwell, "A procedure for desiging exact reconstruction fiolter banks for tree structured subband coders," in *Proc. IEEE ICASSP*, (San Diego, CA), pp. 27.1.1–27.1.4, Mar. 1984.

[18] M. J. T. Smith and T. P. Barnwell, "Exact reconstruction techniques for tree-structured subband coders," *IEEE Transactions on Acoust. Speech Signal Process.*, vol. ASSP-34, pp. 434–441, June 1986.

[19] M. Vetterli, "Splitting a signal into subband channels allowing perfect reconstruction," in *Proc. IASTED Conf. Appl. Signal Processing*, (Paris, France), June 1985.

[20] M. Vetterli, "Filter banks allowing perfect reconstruction," *Signal Processing*, vol. 10, pp. 219–244, Apr. 1986.

[21] J. W. Woods and S. O'Neal, "Subband coding of images," *IEEE Transactions on Acoust. Speech Signal Process.*, vol. 34, pp. 1278–1288, Oct. 1986.

[22] J. W. Woods, *Subband Image coding.* Boston, MA: Kluwer Academic, 1991.

[23] J. Kovačević and W. Sweldens, "Interpolating filter banks and wavelets in arbitrary dimensions," tech. rep., Lucent Technologies, Murray Hill, NJ, 1997.

[24] M. Unser, "Approximation power of biorthogonal wavelet expansions," *IEEE Transactions on Signal Processing*, vol. 44, pp. 519–527, Mar. 1996.

[25] O. Rioul, "Regular wavelets: a discrete-time approach," *IEEE Transactions on Signal Processing*, vol. 41, pp. 3572–3579, Dec. 1993.

[26] M. Antonini, M. Barlaud, and P. Mathieu, "Image coding using wavelet transform," *IEEE Transactions on Image Processing*, vol. 1, pp. 205–220, Apr. 1992.

[27] M. Vetterli and C. Herley, "Wavelets and filter banks: Theory and design," *IEEE Transactions on Acoust. Speech Signal Process.*, vol. 40, no. 9, pp. 2207–2232, 1992.

[28] J. Villasenor, B. Belzer, and J. Liao, "Wavelet filter evaluation for image compression," *IEEE Transactions on image processing*, vol. 2, pp. 1053–1060, Aug. 1995.

[29] G. Deslauriers and S. Dubuc, "Symmetric iterative interpolation processes," *Constructive Approximation*, vol. 5, no. 1, pp. 49–68, 1989.

[30] W. Sweldens, "The lifting scheme: A new philosophy in biorthogonal wavelet constructions," in *Wavelet Applications in Signal and Image Processing III* (A. F. Laine and M. Unser, eds.), pp. 68–79, Proc. SPIE 2569, 1995.

[31] M. Tsai, J. Villasenor, and F. Chen, "Stack-run image coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, pp. 519–521, Oct. 1996.

[32] ftp://eceserv0.ece.wisc.edu/pub/nguyen/SOFTWARE/UIFBD.

[33] I. Balasingham and T. A. Ramstad, "On the relevance of the regularity constraint in subband image coding," in *Proc. Asilomar Conference on Signals, Systems, and Computers*, (Pacific Grove), 1997.

[34] M. T. J. Smith and S. L. Eddins, "Analysis/synthesis techniques for subband image coding," *IEEE Transactions on Acoust. Speech Signal Process.*, pp. 1446–1456, Aug. 1991.

[35] C. M. Brislawn, "Classification of nonexpansive symmetric extension transforms for multirate filter banks," *Applied and Comp. Harmonic Analysis*, vol. 3, pp. 337–357, 1996.

[36] C. Herley and M. Vetterli, "Orthogonal time-varying filter banks and wavelets," in *Proc. IEEE ISCAS*, vol. 1, pp. 391–394, May 1993.

[37] C. Herley, "Boundary filters for finite-length signals and time-varying filter banks," *IEEE Trans. Circuits and Systems II*, vol. 42, pp. 102–114, Feb. 1995.

[38] W. Sweldens and P. Schröder, "Building your own wavelets at home," Tech. Rep. 1995:5, Industrial Mathematics Initiative, Mathematics Department, University of South Carolina, 1995.

[39] S. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 674–693, July 1989.

[40] H. Gish and J. N. Pierce, "Asymptotically efficient quantizing," *IEEE Transactions on Information Theory*, vol. IT-14, pp. 676–683, Sept. 1968.

[41] N. Farvardin and J. W. Modestino, "Optimum quantizer performance for a class of non-Gaussian memoryless sources," *IEEE Transactions on Information Theory*, vol. 30, pp. 485–497, May 1984.

[42] D. Taubman and A. Zakhor, "Multirate 3-D subband coding of video," *IEEE Trans. Image Proc.*, vol. 3, Sept. 1994.

[43] T. C. Bell, J. G. Cleary, and I. H. Witten, *Text Compression*. Englewood Cliffs, New Jersey: Prentice Hall, 1990.

[44] D. L. Duttweiler and C. Chamzas, "Probability estimation in arithmetic and adaptive-Huffman entropy coders," *IEEE Transactions on Image Processing*, vol. 4, pp. 237–246, Mar. 1995.

[45] X. Wu, "Lossless compression of continuous-tone images via context selection, quantization, and modeling," *IEEE Transactions on Image Processing*, vol. 6, pp. 656–664, May 1997.

[46] X. Wu, "High-order context modeling and embedded conditional entropy coding of wavelet coefficients for image compression," in *Proc. of 31st Asilomar Conf. on Signals, Systems, and Computers*, (Pacific Grove, CA), Nov. 1997.

[47] A. Zandi, J. D. Allen, E. L. Schwartz, and M. Boliek, "CREW: compression by reversible embedded wavelets," in *Proc. Data Compression Conference*, (Snowbird, UT), pp. 212–221, 1995.

[48] C. Chrysafis and A. Ortega, "Efficient context based entropy coding for lossy wavelet image compression," in *Proc. Data Compression Conference*, (Snowbird, Utah), pp. 241–250, 1997.

[49] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Transactions on Acoust. Speech Signal Process.*, vol. 36, pp. 1445–1453, Sept. 1988.

[50] P. Moulin, "A multiscale relaxation algorithm for SNR maximization in nonorthogonal subband coding," *IEEE Transactions on Image Processing*, vol. 4, pp. 1269–1281, Sept. 1995.

[51] J. W. Woods and T. Naveen, "A filter based allocation scheme for subband compression of HDTV," *IEEE Transactions on Image Processing*, vol. IP-1, pp. 436–440, July 1992.

[52] B. A. Wandell, *Foundations of Vision*. Sunderland, MA: Sinauer Associates, 1995.

[53] A. B. Watson, G. Y. Yang, J. A. Soloman, and J. Villasenor, "Visual thresholds for wavelet quantization error," in *Proceedings of the SPIE*, vol. 2657, pp. 382–392, 1996.

[54] S. Mallat and F. Falzon, "Analysis of low bit image transform coding," *IEEE Transactions on Signal Processing*, Apr. 1998.

[55] R. A. DeVore, B. Jawerth, and B. J. Lucier, "Image compression through wavelet transform coding," *IEEE Transactions on Information Theory*, vol. 38, pp. 719–746, Mar. 1992.

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

[57] M. Crouse and K. Ramchandran, "Joint thresholding and quantizer selection for decoder-compatible baseline JPEG," in *Proc. IEEE ICASSP*, May 1995.

[58] G. M. Davis, "The wavelet image compression construction kit." http://www.cs.dartmouth.edu/∼gdavis/wavelet/wavelet.html.

[59] J. Shapiro, "Embedded image coding using zero-trees of wavelet coefficients," *IEEE Transactions on Signal Processing*, vol. 41, pp. 3445–3462, Dec. 1993.

[60] A. Said and W. A. Pearlman, "An image multiresolution represenation for lossless and lossy compression," *IEEE Transactions on Image Processing*, vol. 5, pp. 1303–1310, Sept. 1996.

[61] Z. Xiong, K. Ramchandran, and M. T. Orchard, "Space-frequency quantization for wavelet image coding," *IEEE Transactions on Image Processing*, vol. 6, pp. 677–693, May 1997.

[62] K. Ramchandran and M. Vetterli, "Best wavelet packet bases in a rate-distortion sense," *IEEE Transactions on Image Processing*, vol. 2, no. 2, pp. 160–175, 1992.

[63] C. Herley, Z. Xiong, K. Ramchandran, and M. T. Orchard, "Joint space-frequency segmentation using balanced wavelet packet trees for least-cost image representation," *IEEE Transactions on Image Processing*, Sept. 1997.

[64] Z. Xiong, K. Ramchandran, and M. Orchard, "Wavelet packets image coding using space-frequency quantization," *IEEE Transactions on Image Processing*, vol. 7, pp. 892–898, June 1998.

[65] R. L. Joshi, H. Jafarkhani, J. H. Kasner, T. R. Fisher, N. Farvardin, M. W. Marcellin, and R. H. Bamberger, "Comparison of different methods of

classification in subband coding of images," *IEEE Transactions on Image Processing*, vol. 6, pp. 1473–1486, Nov. 1997.

[66] Z. Xiong and X. Wu, "Wavelet image coding using trellis coded space-frequency quantization," in *Proc. IEEE Multimedia Signal Processing Workshop*, (Los Angeles, CA), Dec. 1998.

[67] S. M. LoPresto, K. Ramchandran, and M. T. Orchard, "Image coding based on mixture modeling of wavelet coefficients and a fast estimation-quantization framework," in *Proc. Data Compression Conference*, (Snowbird, Utah), pp. 221–230, 1997.

[68] A. S. Lewis and G. Knowles, "Image compression using the 2-d wavelet transform," *IEEE Transactions on Image Processing*, vol. 1, pp. 244–250, Apr. 1992.

[69] G. M. Davis and S. Chawla, "Image coding using optimized significance tree quantization," in *Proc. Data Compression Conference* (J. A. Storer and M. Cohn, eds.), pp. 387–396, Mar. 1997.

[70] R. R. Coifman and M. V. Wickerhauser, "Entropy based algorithms for best basis selection," *IEEE Transactions on Information Theory*, vol. 32, pp. 712–718, Mar. 1992.

[71] R. R. Coifman and Y. Meyer, "Nouvelles bases orthonormées de $l^2(\mathbf{r})$ ayant la structure du système de Walsh," Tech. Rep. Preprint, Department of Mathematics, Yale University, 1989.

[72] M. V. Wickerhauser, *Adapted Wavelet Analysis from Theory to Software*. Wellesley, MA: A. K. Peters, 1994.

[73] C. J. I. Services, *WSQ Gray-Scale Fingerprint Image Compression Specification (ver. 2.0)*. Federal Bureau of Investigation, Feb. 1993.

[74] C. Herley, J. Kovačević, K. Ramchandran, and M. Vetterli, "Tilings of the time-frequency plane: Construction of arbitrary orthogonal bases and fast tiling algorithms," *IEEE Transactions on Signal Processing*, vol. 41, pp. 3341–3359, Dec. 1993.

[75] J. R. Smith and S. F. Chang, "Frequency and spatially adaptive wavelet packets," in *Proc. IEEE ICASSP*, May 1995.

[76] M. W. Marcellin and T. R. Fischer, "Trellis coded quantization of memoryless and Gauss-Markov sources," *IEEE Transactions on Communications*, vol. 38, pp. 82–93, Jan. 1990.

[77] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Transactions on Information Theory*, vol. IT-28, pp. 55–67, Jan. 1982.

[78] M. W. Marcellin, "On entropy-constrained trellis coded quantization," *IEEE Transactions on Communications*, vol. 42, pp. 14–16, Jan. 1994.

[79] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Entropy-constrained vector quantization," *IEEE Transactions on Information Theory*, vol. 37, pp. 31–42, Jan. 1989.

[80] W. H. Chen and C. H. Smith, "Adaptive coding of monochrome and color images," *IEEE Transactions on Communications*, vol. COM-25, pp. 1285–1292, Nov. 1977.

[81] N. S. Jayant and P. Noll, *Digital Coding of waveforms*. Englewood Cliffs, NJ: Prentice-Hall, 1984.

[82] M. Eyuboglu and J. G. D. Forney, "Lattice and trellis quantization with lattice- and trellis-bounded codebooks–high-rate theory for memoryless sources," *IEEE Transactions on Information Theory*, vol. IT-39, pp. 46–59, Jan. 1993.

[83] P. G. Sherwood and K. Zeger, "Progressive image coding for noisy channels," *IEEE Signal Processing Letters*, vol. 4, pp. 189–191, July 1997.

[84] J. Lu, A. Nosratinia, and B. Aazhang, "Progressive source-channel coding of images over bursty error channels," in *Proc. IEEE Int. Conf. Image Proc. (ICIP)*, (Chicago), Oct. 1998.

[85] S. L. Regunathan, K. Rose, and S. Gadkari, "Multimode image coding for noisy channels," in *Proc. Data Compression Conference*, (Snowbird, UT), pp. 82–90, Mar. 1997.

[86] J. Garcia-Frias and J. D. Villasenor, "An analytical treatment of channel-induced distortion in run length coded image subbands," in *Proc. Data Compression Conference*, (Snowbird, UT), pp. 52–61, Mar. 1997.

[87] P. Topiwala, *Wavelet image and video compression*. Boston, MA: Eds, Kluwer, 1998.

[88] A. Nosratinia and M. T. Orchard, "A multi-resolution framework for backward motion compensation," in *Proc. SPIE Symposium on Electronic Imaging*, (San Jose, CA), Feb. 1995.