An Integrated Framework for Adaptive Subband Image Coding

Vladimir Pavlovic, *Member, IEEE,* Pierre Moulin, *Senior Member, IEEE,* and Kannan Ramchandran, *Senior Member, IEEE*

***Abstract*—Recent work on filter banks and related expansions has revealed an interesting insight: Different filter bank trees can be regarded as different ways of constructing orthonormal bases for linear signal expansion. In particular, fast algorithms for finding best bases in an operational rate-distortion (R/D) sense have been successfully used in image coding. Independently of this work, recent research has also explored the design of filter banks that optimize energy compaction for a single signal or a class of signals. In this paper, we integrate these two different but complementary approaches to best-basis design and propose a coding paradigm in which subband filters, tree structure, and quantizers are chosen to optimize R/D performance. These coder attributes represent side information. They are selected from a codebook designed off-line from training data, using R/D as the design criterion. This approach provides a rational framework in which to explore alternatives to empirical design of filter banks, quantizers, and other coding parameters. The on-line coding algorithm is a relatively simple extension of current R/D-optimal coding algorithms that operate with fixed filter banks and empirically designed quantizer codebooks. In particular, it is shown that selection of the best adapted filter bank from the codebook is computationally elementary.**

***Index Terms*— Adaptive coding, best basis methods, filter banks, image compression, rate-distortion methods, subband coding, vector quantization.**

I. Introduction

TRANSFORM coding has become the *de facto* standard for image and video compression. It is based on the principle that a (linear) transformed version of a given image is often easier to compress (i.e., has better energy compaction and decorrelation properties) than the original signal. The traditional approach has been to use a fixed transform $\mathbf{A}$ (e.g., the discrete cosine transform, the discrete wavelet transform, etc.). Although this may suffice for fixed classes of signals that are well suited in some sense (e.g., in statistical time-frequency characterization) to the fixed transform $\mathbf{A}$, it is limiting when dealing with arbitrary classes of signals having unknown or time-varying characteristics. For example, for image or image segments having high-frequency stationary components, the wavelet transform, which has good frequency selectivity at lower frequencies and good spatial localization at higher frequencies, is a bad fit. This has motivated alternative approaches that are adaptive in their representation and more robust in dealing with a large class of signals of unknown characteristics. The goal is then to make the transform signal-adaptive, i.e., to make $\mathbf{A}$ vary with the signal x. This leads to a search for the "optimal" $\mathbf{A}^*$

$$\mathbf{A}^* = \text{argmin}_{\mathbf{A}}\, \text{Cost}(\mathbf{A}x). \tag{1}$$

for some specified cost function. While minimizing (1) over "all possible" $\mathbf{A}$ is infeasible, choosing a large but finite library of $\mathbf{A}$'s that can be searched efficiently would make finding the optimal $\mathbf{A}^*$ both feasible and desirable.

Recently, wavelets and filter bank theory, along with their generalizations like (adaptive) wavelet packets, have appeared as alternatives to the classic Fourier expansions [1]. An interesting insight to emerge from the work on filter banks and related expansions is that different filter bank trees can be regarded as different ways of constructing signal expansion bases. These trees (termed wavelet packets in [2]) represent a huge library of orthonormal bases having rich space-frequency diversity with easy-to-search capability thanks to the tree structure. Despite the tree structure, the number of library entries is huge, e.g., a depth-5 two-dimensional (2-D) wavelet packet decomposition has a library of 5.6×10^{78} bases! This paradigm, whose main strength is its ability to be signal-adaptive without needing explicit training models, has led to an exciting new area of research on adaptive signal decompositions for compression using wavelet packets and was originally introduced in [2]. The idea is to decompose a discrete signal using all possible wavelet packet bases of a given wavelet kernel and then to find the "best" wavelet packet basis. For signal and image coding, a fast algorithm for finding the best basis in an operational rate-distortion (R/D) sense, i.e., to find a combination of best basis $\mathbf{A}$ and best set of quantizers $\mathbf{Q}$, has been introduced in [3]. In this case, the "optimal" choice is

$$(\mathbf{Q}^*, \mathbf{A}^*) = \text{argmin}_{\mathbf{Q},\,\mathbf{A}} \text{Cost}(\mathbf{Q}\mathbf{A}x) \tag{2}$$

where $\text{Cost}(\mathbf{Q}\mathbf{A}x)$ represents the operational R/D cost of the quantized, transformed image data $\mathbf{Q}\mathbf{A}x$. This approach assumes a fixed overhead cost for encoding the index of the basis $\mathbf{A}$ and the set of quantizers $\mathbf{Q}$. The algorithm prunes a complete tree, signifying the entire library of admissible wavelet packet bases, into that best basis subtree that minimizes the global distortion for a given coding bit budget or

Manuscript received February 26, 1997; revised August 26, 1998. This work was supported by the National Science Foundation under Grants MIP-97-07633 and NSF MIP 97-073181. The associate editor coordinating the review of this paper and approving it for publication was Prof. Stéphane G. Mallat.

V. Pavlovic is with the Cambridge Research Laboratory, Compaq Computer Corp., Cambridge, MA 02139 USA (e-mail: vladimir@crl.dec.com).

P. Moulin and K. Ramchandran are with the University of Illinois, Beckman Institute and Department of Electrical and Computer Engineering, Urbana, IL 61801 USA (e-mail: moulin@ifp.uiuc.edu; kannan@ifp.uiuc.edu).

Publisher Item Identifier S 1053-587X(99)02142-X.

conversely that minimizes the total coding bit rate for a target quality. This algorithm has been subsequently generalized in [4] and [5] to provide for spatial adaptation in addition to frequency adaptation. Such expansions give rise to arbitrary orthonormal tilings of the time–frequency plane.

Independently of these developments, recent efforts have led to the design of orthogonal subband filter banks that are optimally adapted to input signal statistics, in the sense that they maximize the energy compaction of the two-channel filter bank [6]–[10]. The original paper [6] demonstrated some of the advantages of this approach in multiresolution image coding. Until recently, an obstacle to the use of such adaptive methods was the limited performance and substantial numerical complexity of the optimization algorithms involved. However, [9] and [10] showed that the optimization problem may be reformulated in terms of the *product filter* $P(f) = |H(f)|^2$ associated with the lowpass filter $H(f)$ and becomes a linear semi-infinite programming problem. This formulation provides a framework for analyzing the performance of signal-adapted filter banks, as well as fast and reliable algorithms for computing globally optimal filter banks. The filter $H(f)$ is obtained by spectral factorization of the optimal product filter. The solution is not unique; a typical choice is the minimum-phase solution. The coding gain for filter banks adapted to lowpass processes is marginally higher than that for conventional filter banks, but significant improvements have been obtained with image textures [6], [10].

The authors in [2]–[5] on the one hand, and [6]–[10], on the other hand, approach the problem of designing best bases using filter banks from different but complementary angles. Our main objective in this paper is to integrate these paradigms and to propose a new coding framework in which the subband filters, tree structure, and quantizers are jointly designed adaptively. In this sense, our proposed paradigm can be viewed as a generalization of the adaptive tree-structured best basis framework of [2]–[5] to include the freedom to change the filters at each node of the subband tree. Additionally, as will be detailed later, due to the information gleaned during the training phase, our paradigm results in potentially much improved on-line speed due to two factors. First, the candidate set of filter bank tree structures over which to search for the best basis can be smaller than the traditionally considered set of all pruned subtrees of a full tree of (sufficiently) large depth. Second, the list of candidate quantizer options can be potentially much trimmer than an *ad hoc* set of needlessly large number of choices.

Despite these computational advantages, we wish to emphasize that the main contribution of our work, as we see it, is the formulation of a systematic framework in which to study fundamental problems of selection and coding of subband coder parameters. The particular optimization algorithms used in this paper are only a first step toward an extremely ambitious goal, and future research on optimization techniques should further improve practical performance of the coder.

A. Notation

Filter bank and quantizer parameters are represented by vectors FB and Q, respectively. For clarity of the exposition, we restrict our attention to uniform scalar quantizers, in which case, Q is a single parameter, the quantizer step size. However, the concepts presented here apply to more complex quantizers, such as nonuniform scalar quantizers and vector quantizers. The tree structure may be described by the splitting decision $s \in \{0, 1\}$ at each node n of the tree. Our transform is described by the collection of pairs $\{(FB^{(n)}, s^{(n)})\}$; likewise, the quantizers are described by a collection $\{Q^{(n)}\}$. A particular choice of transform and quantizers is viewed as a codeword c; the collection of all codewords is a codebook $\mathcal{C}$. We use the symbol $l(\cdot)$ to denote codeword length.

B. On Codebook Complexity

Our proposed generalization of the best-basis framework of [2]–[5] raises two important issues: computational complexity and the increased overhead cost for coding applications. Node-split decisions are binary-valued and losslessly encoded. However, quantizer step sizes can be arbitrary positive real numbers, and filter bank parameters belong to a continuum of admissible values as well. Clearly, all these parameters must somehow be quantized, and a fundamental tradeoff arises between accuracy of this quantization (hence, amount of the side information) and adaptation performance. We reformulate the R/D optimization problem (2) to include the overhead cost for encoding the index of $\mathbf{Q}$ and $\mathbf{A}$. The optimal choice for $\mathbf{Q}$ and $\mathbf{A}$ becomes

$$(\mathbf{Q}^*, \mathbf{A}^*) = \operatorname{argmin}_{\mathbf{Q}, \mathbf{A}}[\text{Cost}(\mathbf{Q}, \mathbf{A}) + \text{Cost}(\mathbf{QA}x)]. \quad (3)$$

The fundamental tradeoff between the two terms on the right-hand side of (3) may be dealt with in a number of ways. At a higher level, these may be classified as belonging to nontraining-based versus training-based methods. The former is attractive when statistical priors are explicitly available. One approach is to assume a uniform prior, and assume that the overhead cost is independent of $\mathbf{Q}$ and $\mathbf{A}$. This approach has been used in [3]–[5]. When training data is available, however, training-based frameworks can be more efficient as they "learn the prior" from the training data. This second approach (training-based) is central to the new framework we propose. We briefly enumerate the two popular approaches cited above.

1) Empirical Design [3]–[5]: A prespecified collection of quantizers $\{Q^{(n)}\}$ is assigned to each node. A typical choice in practical image coding applications is to use a discrete set of uniform scalar quantizers, with step sizes equally spaced over a prespecified range. Quantizer indices are encoded using a fixed-length code, implicitly assuming a uniform distribution of the indices.[1] The depth of the tree is typically limited in practice to three or four resolution levels. For example, for a 512×512 image, a quantizer set of 32 choices, and a maximum depth-3 tree, the total overhead for the quantizer and tree description comes to less than 0.001 34 b/pixel, which is certainly negligible for even low bit rate coding applications.

[1] Note that it is the quantizer step-size choice that is coded using a fixed-length code. The quantized symbol stream can be coded using a variable-length entropy code.

The set $\{FB^{(n)}\}$ of possible filter banks should also be discretized, and one may empirically choose the accuracy with which the filter bank parameters are to be represented. For instance, if 32 prespecified candidate filter banks are allowed at each node, the total overhead for the filter bank description is the same as above.

2) Training: One disavantage of the codebook design method above is that the set of quantizers at each node of the tree, the set of filters, and the number of resolution levels in the tree are chosen empirically instead of by optimizing the fundamental tradeoff between Cost($\mathbf{QA}x$) and Cost($\mathbf{Q}$, $\mathbf{A}$) in (2). In particular, uniform quantization and weighting of quantizer and (more importantly) filter bank parameters, is inefficient. The underlying assumption of uniform parameter distribution does not reflect the high likelihood of certain filter banks (such as the classical designs in [1]) relative to other admissible but pathological choices. This observation motivates an alternative approach, namely, optimal design of the codebook with respect to some broad ensemble of image sources. Training data are used for learning the statistics of this ensemble and designing the codebook. Codebook design might be computationally intensive but is performed *off-line*, unlike the actual test image to be encoded *on-line*.

Off-line optimization of the filters (into filter classes based on statistical training data) is attractive not only from an overhead cost but also from a computational complexity viewpoint as on-line operation reduces to an elementary low-complexity classification task rather than a complicated filter design operation (see Section III).

C. Overview of the Approach

Each element c of the codebook $\mathcal{C}$ contains attributes of the transform and the quantizers. The codeword is viewed as the first part of a code for encoding the test image, and the image data encoded with respect to these attributes are viewed as the second part of the code. This simple but powerful paradigm has been used for constructing coders that perform well on a variety of sources with (partially) unknown statistics [11], [12].

The total number of bits for encoding the test image I is the sum of the lengths $l(c)$ and $l(I|c)$ for the first and second parts of the code, respectively. Denote by $\hat{I}$ the image reconstructed by the decoder. We define the operational distortion $D(I|c)$ as the mean-squared value of the reconstruction error $\hat{I} - I$. In order to apply our formulation to various R/D tradeoffs, we take the classical approach of minimizing the Lagrangian cost function

$$J(c|I) \triangleq D(I|c) + \lambda[l(c) + l(I|c)] \tag{4}$$

where λ is the Lagrangian multiplier that trades off rate against distortion. This criterion is of the general form (3). The Lagrangian optimization problem (4) arises when the encoder needs to mimimize distortion subject to a rate constraint, or vice-versa: λ represents the slope at the operating point on the R/D curve.[2] The first stage of the encoder minimizes (4) over $c \in \mathcal{C}$, producing an optimal codeword $\hat{c}(I)$. We solve this problem using a fast optimization algorithm to be described in Section III.

The design of the codebook is described in Section II. The goal is to construct this codebook in a R/D optimal fashion over some broad ensemble $\mathcal{I}$ of image sources. More specifically, it is desired to minimize the average cost

$$J_1(\mathcal{C}) \triangleq \int_{\mathcal{I}} J(\hat{c}(I)|I)P(dI)$$

where $P(dI)$ is the underlying probability distribution over $\mathcal{I}$. While $P(dI)$ is unknown, a *training set* $\mathcal{T}$ of images representative of $\mathcal{I}$ is available; therefore, the actual codebook design problem consists of minimizing the empirical average

$$\hat{J}_1(\mathcal{C}) \triangleq \frac{1}{|\mathcal{T}|} \sum_{I \in \mathcal{T}} J(\hat{c}(I)|I). \tag{5}$$

The solution consists of partitioning the ensemble $\mathcal{I}$ into appropriate subsets and assigning to each subset a representative transform and quantizer, encoded using a variable-length codeword c. In other words, the codebook design is a classification problem. Our design problem bears obvious similarities to entropy-constrained vector quantizer (ECVQ) design [13], which we briefly describe in the Appendix for completeness. While the spirit of our approach is captured by the above ECVQ codebook design algorithm, in our case, the codewords represent attributes of the transform and quantizers rather than image data, as would be the case in ECVQ. The theoretical foundations for such an approach are described in [11].

Although the approach embodied by (5) is conceptually appealing, its solution appears to be beyond reach, due to the large dimensionality of the set of attributes and nonlinear interactions between these attributes. This difficulty is compounded by the dependency of *each* attribute on *all* image pixels! The optimization problem would be simplified if block transforms were used since in this case, transform/quantizers attributes would depend only on local image blocks [12]. Even so, the optimization problem (5) remains highly nonlinear, and it is necessary to construct an approximation to the optimal solution. We use the theoretical coding gain [1], [13] approximation to (5) for the design of the filter banks. This is a critical approximation that might be questionable at low bit rates. We can retain the original formulation (5) for the design of quantizers.

A possible generalization of our approach is outlined in Section IV. Numerical results illustrating important aspects of the design are presented in Section V, and conclusions are presented in Section IV.

II. Codebook Design

The algorithm uses a training set $\mathcal{T}$ made of images assumed to be representative of the ensemble $\mathcal{I}$ of images of interest. Our codebook design algorithm is motivated by a number of practical considerations. We first design the filters and then the quantizers. While our filter design is independent of rate constraints, the design of the tree and quantizers is very

[2] As has been pointed out in [3], there is a convex relationship between the target R and the target λ, which results in a fast bisection-type algorithm to relate the two constraints.

much dependent on the filters. The primary motivation for our sequential design is that the filters are meant to capture the statistical properties of the input, and their design is guided by the theoretical coding gain cost function that is independent of specific coding bit budget constraints. We also note that even though the inclusion of a R/D design constraint in the filters would clearly be a reasonable alternative to our approach, the resulting design problem currently appears to be intractable; it would also heavily complicate the storage requirements as well as the complexity of design during both training and coding phases. This arises from the nonlinear form of (5) as well as from the existence of coupled dependencies between *all* filter banks and quantizers.

A. Filter Bank Design

Define the *theoretical coding gain* for the two-channel filter bank FB applied to input I as

$$G(FB|I) = \frac{\left(\sigma_h^2 + \sigma_g^2\right)/2}{\left(\sigma_h^2 \sigma_g^2\right)^{1/2}} \tag{6}$$

where σ_h^2 and σ_g^2 are the energies of the signals in the two channels. This definition in terms of energies instead of variances is adopted for the sake of consistency with our operational R/D approach. Assuming the simple model $D(R) \propto \sigma^2 2^{-2R}$ for the R/D characteristic in each band, high-rate quantization, and optimal bit allocation between subbands, the theoretical coding gain (6) represents the ratio of distortions produced by the subband coder and a PCM coder operating at the same bit rate. The quantity $\frac{1}{2}\log_2 G(FB|I)$ represents the bit rate reduction over PCM, when both coders operate at the same distortion level [13].

Computation of Optimal Filter Banks: Due to the orthogonality of the filter bank, $G(FB|I)$ may be maximized by maximizing σ_h^2 over FB. Although this is a highly nonlinear optimization problem with a large number of local extrema, the theoretical coding gain solely depends on the product filter

$$\begin{aligned} P(f) &= |H(f)|^2 \\ &= 1 + 2\sum_n a_n \cos(2\pi(2n+1)f) \geq 0 \\ &\quad 0 \leq f \leq 0.5 \end{aligned} \tag{7}$$

when the following assumptions are satisfied [9], [10].

Assumption 1: The input signal is extended beyond its boundaries using periodic extensions.

Assumption 2: The energies of the signal before and after decimation of the output of $H(f)$ are identical.

Under these assumptions, the optimization problem may be formulated in terms of $P(f)$ and becomes a semi-infinite linear optimization problem with linear objective function

$$\sigma_h^2 = r_0/2 + \sum_n a_n r_n \tag{8}$$

and infinitely many linear inequalities (7). In this formulation, r_n are the empirical correlation coefficients for the input. The optimization problem may be solved using fast algorithms based on a discretization of the frequency interval [0, 0.5] and a standard simplex algorithm.

On the Assumptions Used: Although Assumption 2 is approximately satisfied for stationary or slowly varying signals, it remains to be seen whether the assumption holds for real-world images, at various resolution levels. This issue is examined in Section V. Another interesting point is that while maximization of σ_h^2 implies maximization of the coding gain (6) under the various assumptions above, maximization of σ_h^2 is still optimal in the R/D sense when some of these assumptions are not satisfied. Consider, for instance, scalar quantizers with operational R/D functions of the form $D = \sigma^2 \overline{D}(R)$, where $\overline{D}(\cdot)$ is independent of σ^2 and is *strictly monotonically decreasing* (but not necessarily convex) over the positive half real line. This model includes the standard model $\overline{D}(R) = \gamma 2^{-2R}$ as a special case. Property 1 below shows that the filter banks that maximize σ_h^2 are still optimal in the R/D sense for any strictly montonic $\overline{D}(R)$ and for any bit allocation. This property applies to arbitrary classes $\mathcal{F}$ of orthonormal filter banks, including constrained-length filter banks.

Property 1: Let $\mathcal{F}_{opt}$ be the set of filter banks that maximize σ_h^2 (or, equivalently, minimize σ_g^2) over some subset $\mathcal{F}$ of the set of orthonormal filter banks. Then, for a fixed bit allocation $\{R_h, R_g\}$ where $R_h > R_g$, the set of minimizers of the overall distortion function $D = \frac{1}{2}\left[\sigma_h^2 \overline{D}(R_h) + \sigma_g^2 \overline{D}(R_g)\right]$ is identical to $\mathcal{F}_{opt}$. For $R_h = R_g$, any filter bank in $\mathcal{F}$ is optimal.

Proof: For $R_h = R_g$, we have $D = \frac{1}{2}(\sigma_h^2 + \sigma_g^2)\overline{D}(R_h) = \sigma_x^2 \overline{D}(R_h)$ for all filter banks in $\mathcal{F}$. For $R_h > R_g$, let D_* be the distortion attained using a filter bank in $\mathcal{F}_{opt}$, and let σ_{h*}^2 and σ_{g*}^2 be the corresponding channel variances. We have $\overline{D}(R_h) - \overline{D}(R_g) < 0$ and $\sigma_h^2 - \sigma_{h*}^2 = -(\sigma_g^2 - \sigma_{g*}^2) \leq 0$. Hence, $D = D_* + \frac{1}{2}(\sigma_h^2 - \sigma_{h*}^2)(\overline{D}(R_h) - \overline{D}(R_g)) \geq D_*$, with equality if and only if $\sigma_h^2 = \sigma_{h*}^2$.

Optimization of Cascaded Filter Banks: The practical linear optimization algorithms above (as well as the nonlinear optimization algorithms in [6] and others) apply to optimization of *two-channel* filter banks, and extending these methods to the case of cascaded filter banks appears to be a formidable problem. The approach recommended in [6] is to successively optimize filter banks, starting from the root node of the tree, using the "local" coding-gain cost function above rather than a global measure of R/D performance for the entire tree-structured subband coder. With this approach, the filters designed at a given node of the tree depend only on the filters designed at the ancestor nodes. Despite its suboptimality, the "local" optimization approach is often used based on the assumption that good energy compaction at local nodes should lead to good coding performance.

Filter Codebook Design: In the filter design phase, we visit nodes in a top-down fashion and design a separate codebook $\mathcal{F}^{(n)} = \{FB_i^{(n)}\}$ for each node n. A training set $\mathcal{T}^{(n)}$ for node n is obtained by filtering and decimating images from the training set $\mathcal{T}^{(\gamma n)}$ for the parent node γn, using the optimal filter assignment from $\mathcal{F}^{(\gamma n)}$. (The root node uses the original training set $\mathcal{T}$.) Partition the training set $\mathcal{T}^{(n)}$ into classes $\mathcal{T}_i^{(n)}$ (or class i, for short) of images that use the same filter bank $FB_i^{(n)}$. The codebook $\mathcal{F}^{(n)}$ is designed to minimize the

theoretical bit rate that is normalized for convenience relative to a PCM coder operating at the same distortion level

$$\sum_i \sum_{t \in \mathcal{T}_i^{(n)}} \left\{ -\frac{N^{(n)}}{2} \log_2 G\left(FB_i^{(n)} \middle| t\right) + l\left(FB_i^{(n)}\right) \right\} \quad (9)$$

where

$N^{(n)}$	size of the image at node n;
$G(FB_i^{(n)}\|t)$	given by (6);
$l(FB_i^{(n)})$	codelength for $FB_i^{(n)}$.

We use the *theoretical codelength* $l(FB_i^{(n)}) = -\log_2 p(i)$, where $p(i) = |\mathcal{T}_i^{(n)}|/|\mathcal{T}^{(n)}|$ is the relative size, or popularity, of class i. Under the various assumptions made above, minimizing (9) is equivalent to minimizing (5) over a subset of $\mathcal{C}$ (filter banks at node n). We re-emphasize the sequential (top–down) nature of this design.

Each class i contains images that are assigned the same filter bank $FB_i^{(n)}$. This is not to say that these images have the same, or even similar, correlation structure. As we have observed experimentally, two images may have widely different empirical correlation coefficients r_n yet identical optimal filter banks.

Classification Algorithm: The classification is performed using an algorithm similar to the Lloyd algorithm for ECVQ design [12], [13], choosing an initial (possibly large) set of filter banks and codewords $l(FB_i^{(n)})$, and iterating between three steps:

1) (*Weighted nearest neighbor condition*) $\forall t \in \mathcal{T}^{(n)}$: Find the class assignment that minimizes (9) by evaluation of new possible memberships for each individual image in the training set.
2) Update theoretical codelengths $l(FB_i^{(n)}) = -\log_2 p(i)$ based on current class popularities $p(i)$.
3) (*Centroid condition*) $\forall i$: Compute the optimal $FB_i^{(n)}$ for class i by exhaustive evaluation of all current candidates and selection of the one that minimizes the inner summation in (9). The quantity $G(FB_i^{(n)}|I)$ is obtained from (6), where the energy σ_h^2 is evaluated using (8) for each filter bank candidate.

This algorithm is greedy, and the iterations are stopped when improvements in the cost function fall below a specified threshold. On completion of the iterations, each training sample t is assigned to a class i with attribute $FB_i^{(n)}$, as represented pictorially at the top of Fig. 1. One possible initial set of filter banks would consist of the optimal filter banks adapted to each individual image in the training set plus a few of the standard (nonadapted) filter banks that have been extensively used in the image coding literature, e.g., Daubechies' filter banks [1], [14]. An interesting question is: how large are the classes represented by these nonadapted filter banks on convergence of the codebook design algorithm? This question is answered in Section V.

In addition, note that the computation of the optimal $FB_i^{(n)}$ for class i in Step 3 of the classification algorithm does not require expensive adapted filter design. We use the semi—infinite programming algorithm only in the initialization phase to find the optimal filter banks adapted to each individual image in the training set.

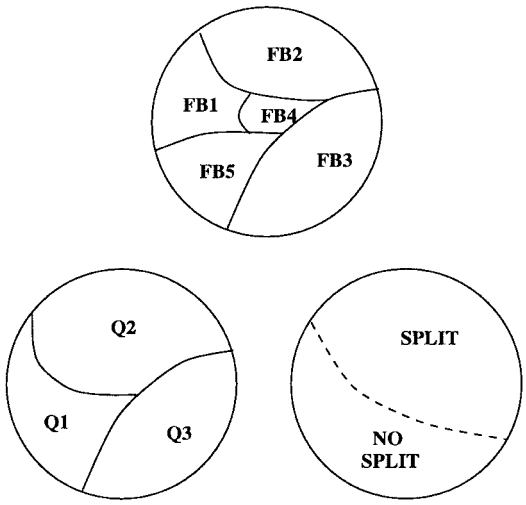


Fig. 1. Classification of filter banks, quantizers, and node-split decisions at a particular node of the subband tree.

B. Tree and Quantizer Design

Having designed the filters, we address the design of quantizers $Q_i^{(n)}$ in the codebook $\mathcal{Q}^{(n)}$. This design is done based on filtered data and, hence, depends on the filter bank codebook designed in Section II-A. The design may be done using standard bit allocation formulas based on the theoretical coding gain, as in [12]. However, it is also feasible to optimize the original cost function (5) directly, exploiting the additivity of (4) over nodes of the tree. We also design weights for the splitting decisions. The design of quantizers and node-split weights is done *jointly*. Each training sample $t \in \mathcal{T}^{(n)}$ is filtered using the assigned filter bank, and the filtered data are viewed as a new training set $\mathcal{T}'^{(n)}$. The design is done using a Lloyd-like greedy iterative algorithm once again, alternating between the assignment of training samples to quantizer classes based on a Lagrangian cost function (Step 1—weighted nearest neighbor condition) and the optimization of the quantizer for samples mapping to the same class (Step 2—centroid condition), with the quantizer weights being updated according to the quantizer's relative popularity from one iteration to the next (Step 3). The algorithm requires an initial (possibly large) set of quantizers and associated codewords. Typically, all quantizers are initially assumed to be equally probable [$l(Q) \equiv$ constant].

1) a) For fixed quantizers and weights, optimize assignment of quantizers to training samples for each tree node using a Lagrangian metric $J^{(n)}(Q) = D^{(n)}(Q) + \lambda(R^{(n)}(Q) + l(Q))$, where $D^{(n)}$ and $R^{(n)}$ are, respectively, the operational distortion and rate associated with quantizer Q at node n, and $l(Q)$ is the codelength for Q.
   b) Optimize the tree structure for each training sample using the single tree algorithm of [3], as shown in Fig. 2. This involves growing the full subband tree to some fixed depth and populating each tree

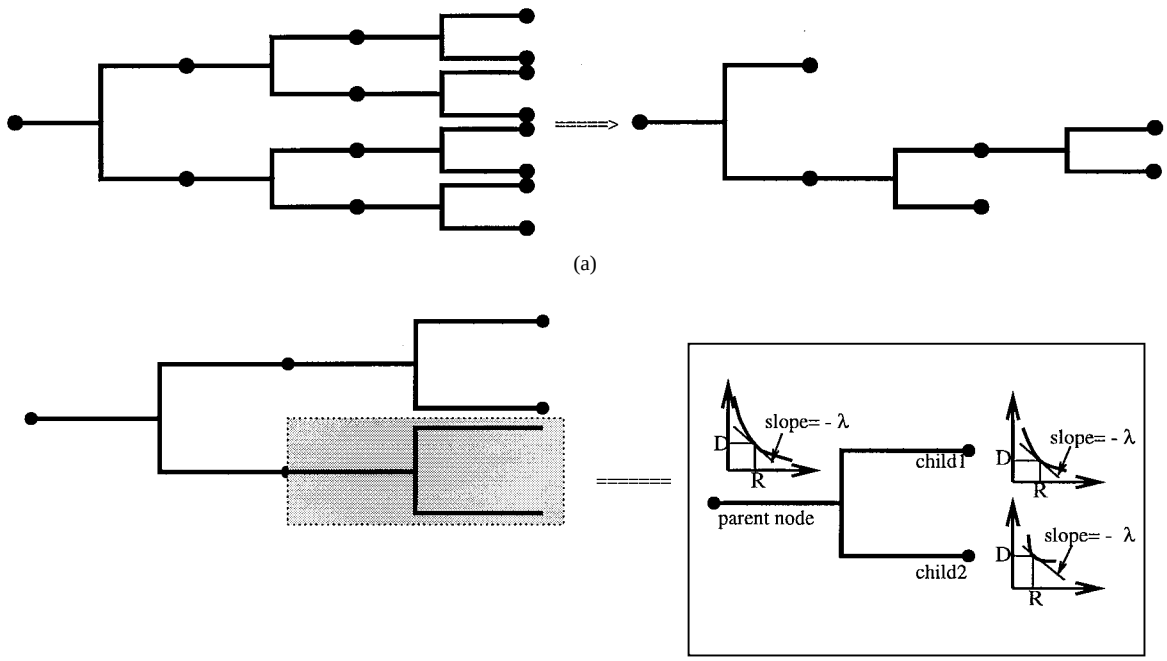


Fig. 2. Single tree algorithm finds the best basis subtree for a given 1-D signal. (a) Algorithm starts from a full tree and prunes back from the leaf nodes to the root node until the best pruned subtree is obtained. (b) At each node, the split-merge decision is made according to the criterion: prune if $J(\text{parent node}) \leq J(\text{child 1}) + J(\text{child 2}) + \lambda(\text{overhead cost for splitting})$.

node with the Lagrangian cost (for a fixed λ) of encoding each associated subband (or the original signal for the root node). A bottom-up tree pruning operation is then performed from the leaves (coarsest scales) toward the root. At each node, a comparison is made between the best Lagrangian cost $J_*^{\text{parent}} = \min_Q J^{\text{parent}}(Q)$ associated with the parent versus the sum of the Lagrangian costs for each child $J_*^{\text{child}} = \min_Q J^{\text{child}}(Q)$ plus λ times the overhead cost for splitting, as measured by the codelengths $l(FB)$ and $l(s)$ for the filter banks and splitting decisions. The single tree algorithm can be summarized as follows.

- Grow a full tree to some fixed depth.
- For a fixed λ, populate each node of the full tree with the best Lagrangian cost $D + \lambda R$.
- Prune the full tree recursively, starting from the leaf nodes.

This results in an optimal node-split classification of training samples for each tree node, given the current codebook $\mathcal{Q}^{(n)}$.

2) Optimize the quantizer vector $Q_i^{(n)}$ for each class i and for each tree node n. Note that the node-split decisions from Step 1(b) will prune the class memberships of training samples at certain deeper nodes.
3) Update the codelengths $l(Q_i^{(n)})$ and $l(s^{(n)})$ of the quantizers and the node-split decisions for each node, and revisit Step 1.

After convergence, the algorithm results in optimized quantizer designs, as well as codewords for the quantizers and the node-split decisions for each node of the tree; see Fig. 1. The primary motivation for including splitting decisions into the algorithm is not to economize on $l(s)$, which is no more than 1 bit/node, but rather to affect quantizer design through the class pruning operation of Step 2.

III. Coding Algorithm

The "on-line" coding algorithm is similar to the dynamic-programming based single tree algorithm of [3]. However, unlike the latter algorithm, which is based on fixed *ad hoc* choices for filter banks and quantizer sets, the proposed algorithm uses optimized sets of filter and quantizer classes (together with their "weights" as measured by their entropies) as designed off-line based on statistical training data; see Section II. Like the single tree algorithm of [3], our algorithm finds the optimal tree-structure (or so-called best basis) and quantizer choice for each tree node jointly in an image-adaptive manner. However, in addition, the proposed algorithm also finds the best filter bank choices (from among the class of filter banks designed off-line) for each tree node. The algorithm proceeds as follows.

1) For the given subband image, find the optimal filter bank (i.e., select optimally from the set of candidates) for successive nodes of the tree using the theoretical coding gain criterion. Note that this is computationally very attractive as the coding gain computation is based only on the input correlation vector at each node of the tree and does not require expensive adapted filter design. Specifically, at each node, the correlation coefficients r_n for the image (see Section II-A) are computed, and the energy σ_h^2 is evaluated using (8) for each filter bank

(each a). The latter step requires only N arithmetic operations per filter bank, where $2N$ is the length of the filters, independently of subband size. This step is computationally straightforward and does not involve any expensive filter design. We then select the filter bank that minimizes the R/D cost

$$-\frac{N^{(n)}}{2}\log_2 G\Big(FB_i^{(n)}|I\Big)+l\Big(FB_i^{(n)}\Big)$$

where $G(FB_i^{(n)}|I)$ is obtained from (6).

2) Populate each node n of the tree with the Lagrangian costs

$$J^{(n)}(Q)=D^{(n)}(Q)+\lambda\Big(R^{(n)}(Q)+l(Q)\Big) \qquad (10)$$

associated with each quantizer $Q\in\mathcal{Q}^{(n)}$. Note that the inclusion of $l(Q)$ in (10) reflects the cost of the quantizer that is based on its relative "popularity" as gleaned during training.

3) Apply the single tree algorithm of [3] as described in the quantizer design phase [see Step 1(b) of Section II-B] to find the optimal quantizers and node-split (pruning) decisions using a dynamic-programming based "bottom-up" approach from the leaves of the tree toward the root. This step results in the jointly optimal combination of tree and quantizer choices for the test image. In combination with the first step, this results in the optimal combination of filters, tree structure, and quantizer choices for the test image.

A novel aspect of our coding algorithm is that it efficiently balances the advantages of off-line training with the flexibility of on-line R/D based optimization. This flexibility helps alleviate susceptibility to mismatches between the statistics of the test data and that of the training data.

The decoupling of the filter bank optimization and that of the tree structure and quantizer is made possible by the choice of the coding gain cost function for the filter bank optimization. A R/D cost function to select the optimal filter bank would make the joint optimization of the filters, tree-structure, and quantizers much more difficult due to the complex interdependencies introduced between different nodes.

A salient advantage of the new approach over the method in [3] is that the on-line coding algorithm is *computationally more efficient* than that in [3]. This apparently surprising advantage occurs because codebook design produces a relatively small number of candidate quantizers, which is the "right" set of quantizer choices to use. This avoids the need to "blindly" consider a needlessly large set of choices, as in [3]. An even greater practical benefit brought about by the off-line training phase is the potentially significant on-line complexity reduction in the tree optimization phase. While the typical approach of [3] is to populate the full subband tree to an *ad hoc* depth and prune it optimally, our proposed framework allows for the potential to start from an optimal subtree rather than the full tree. The optimal subtree is learned during training based on the zero probability of certain tree nodes with respect to the representative training data. Furthermore, the starting tree depth no longer needs to be *ad hoc* but can be learned from training. As an example, when dealing with a class of lowpass processes that favor a logarithmic split of the frequency subband tree (as learned during training), it is unnecessary to start from the computationally expensive fullband tree, as in [3]. In fact, the complexity bottleneck in the on-line algorithm is in the tree population phase, which involves expensive filtering operations. Minimizing this phase is the key to reducing overall on-line complexity. In this regard, if further complexity reduction is desired at the cost of slight suboptimality, it is possible to employ simpler greedy top–down optimization algorithms [15] rather than the optimal bottom–up tree-pruning algorithms advocated earlier.

One drawback of our current method is that it does not exploit spatial inhomogeneity; in the next section, we suggest an extension of the method that might address this limitation.

IV. Extension to Space-Frequency Trees

A natural extension of the adapted-transform paradigm considered in [3] and in this paper consists of allowing the transform $\mathbf{A}$ to be not only frequency adaptive but *spatially adaptive* as well. One of the drawbacks of the frequency-adaptive wavelet packet framework is that although it selects the best tree adapted to a signal, it retains that tree for the entire signal. If the signal is nonstationary, the algorithm will choose a basis that works best "on average" for the whole signal but cannot adapt the tree to different segments of the signal. The double tree structure of [4] alleviates this problem by admitting a dyadic spatial segmentation of wavelet packet (frequency) trees together with a fast tree-pruning algorithm to find the best double tree basis. This structure has been extended to the balanced space-frequency tree (SFT) structure of [16], which represents the most general form of dyadic segmentation in both space and frequency and includes both the best wavelet packet basis and the best double tree basis as special cases. A fast algorithm to find the resulting transform $\mathbf{A}$ represents the best SFT basis for the given signal has been described in [16]. Since the transform $\mathbf{A}$ still admits a tree-structured representation, the codebook design and on-line coding algorithms are, at least conceptually, straightforward extensions of the algorithms presented in Sections II and III. The on-line SFT coding algorithm now evaluates node splits, candidate quantizers, and candidate filter banks at each node of the tree.

V. Experimental Results

We describe some experiments that illustrate the concepts introduced in this paper. The primary goal of these experiments was to gain a better understanding of the nature of the adaptive subband coding process, by examining complexity tradeoffs, in particular, the relation between subband image statistics and adapted filter responses, and the need for adaptation at various levels of the tree. We also report preliminary coding results.

First, we selected a training set assumed to be representative of five types of images:

1) fingerprints;
2) faces;
3) fabrics;

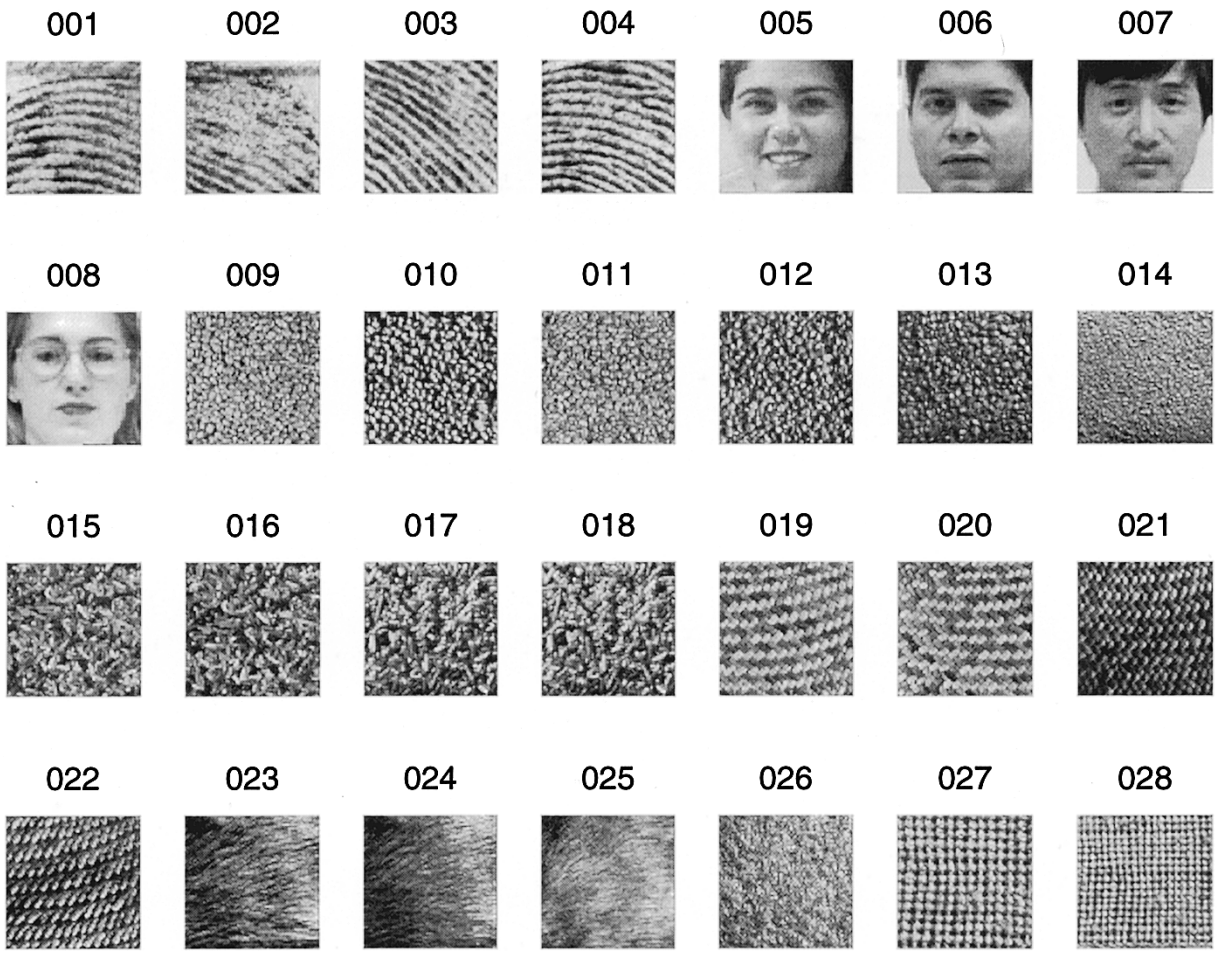


Fig. 3. Training set fingerprints (images 001–004), faces (images 005–008), grains (images 009–014), food (images 015–018), and fabrics (images 019–028).

4) food;
5) grains.

This training set consists of 28 128 $\times$ 128 images and is shown in Fig. 3. The set was chosen in order to highlight the possible advantages of filter bank optimization rather than to provide a typical image set. Fig. 4 depicts our node-indexing scheme in a depth-2 subband quad tree. In our experiments, we used depth-3 trees (with 85 nodes) and eight-tap filters. The individual codebooks for the filter banks, node-split decisions, and quantizers were constructed as described in Section II.

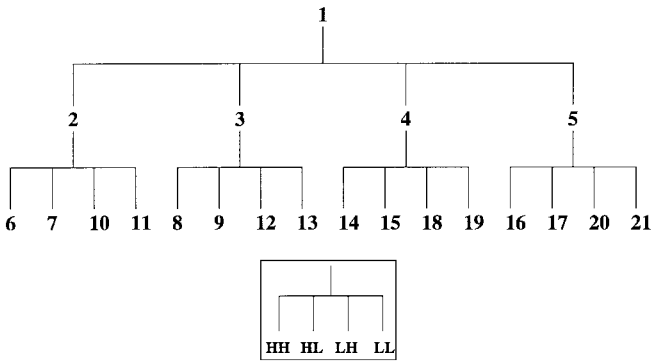


Fig. 4. Indexing of nodes in a depth-2 subband quad tree. Each node has four children ranked according to energy, going from HH (high vertical/high horizontal) to LL (low/low).

A. Adapted Filter Banks

The following operations are performed at each internal node of the subband tree: first, vertical filtering (along columns of the image) and then horizontal filtering of the two vertically filtered images.[3] There are three filter banks per internal node, for a total of 63 filter banks. Each initial codebook consisted of the filter banks optimally adapted to each of the 28 images in the training set, plus Daubechies' eight-tap nonadapted D4 filter bank [14, p. 195]. The optimality (in the theoretical coding gain sense) of the adapted filter banks relies on Assumption 2 in Section II-A. We found this assumption to be reasonably well satisfied, as the mismatch between energies before and after decimation was typically less than 10%. We may expect the initial codebook to consist of 29 different filter banks, but the actual number is lower; the same adapted filter bank is often optimal for multiple images.

The initial set of filter banks was subsequently pruned by the classification algorithm in Section II-A, resulting in a smaller codebook on convergence of the algorithm. The D4 filter banks were pruned out of most codebooks. Fig. 5 shows adapted frequency responses in the high-energy channel. In Fig. 5(a),

[3]Note that this is an arbitrary choice and that switching the order of horizontal and vertical filtering operations would, in general, produce a different codebook.

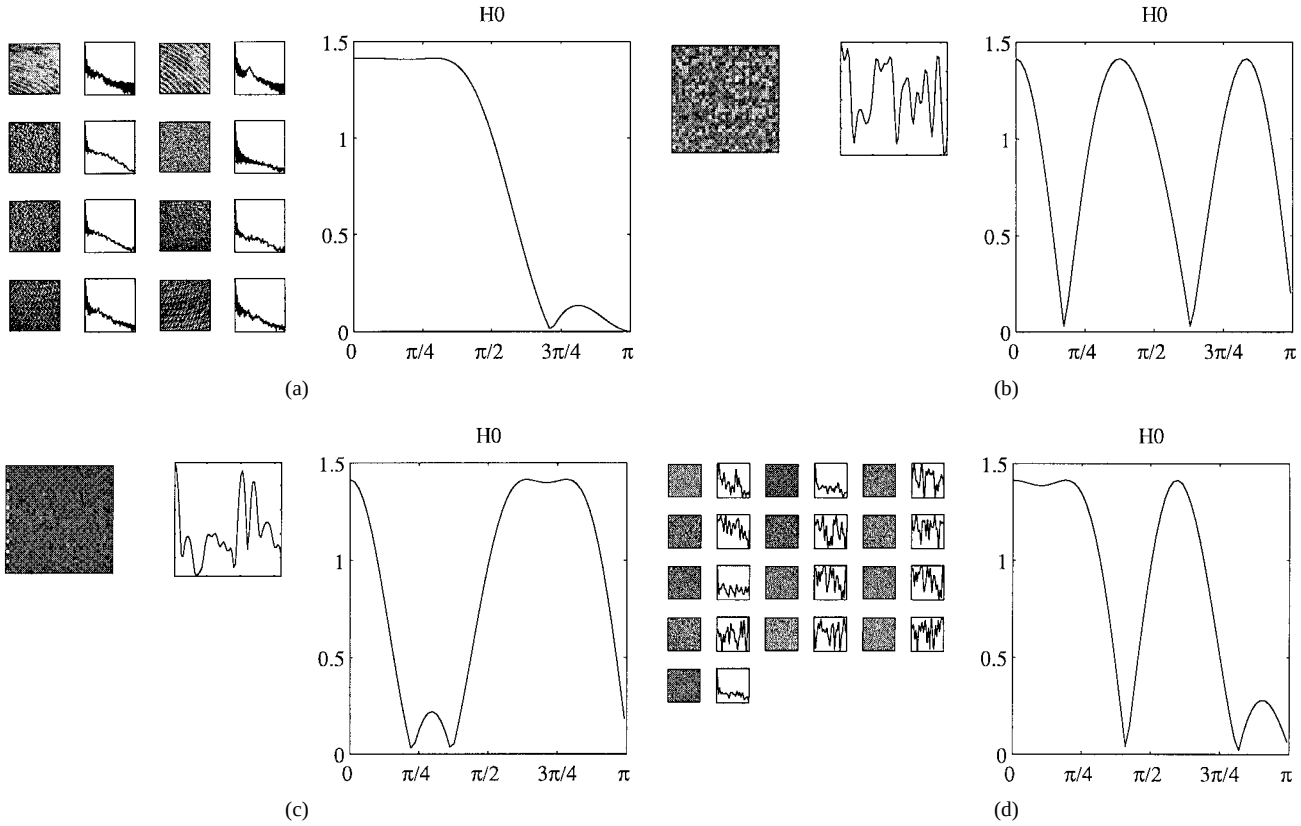


Fig. 5. Class of training images and the corresponding adapted frequency responses. Left side: Classes of images and their estimated log power spectra along columns. Right side: Adapted frequency response. (a) Adapted filter for lowpass processes resembles a conventional frequency-selective lowpass filter. (b) and (c) Adapted filters for exotic processes are also exotic. (d) Processes with seemingly different frequency statistics share the same adapted filter.

eight training images with similar lowpass characteristics share a common lowpass filter. The frequency response of the adapted filter is consistent with the theory in [8], which predicts that the "ideal" (without length constraint) adapted filter is a brickwall lowpass filter with cut-off frequency $\pi/2$ and with the analysis in [10], which predicts that optimal adapted FIR filters have zeros on the unit circle. We routinely observed the presence of frequency-selective filters of the type shown in Fig. 5(a) in codebooks at various nodes of the tree. However, it is also known that adapted filter banks improve only marginally over standard nonadapted filter banks when the input process is lowpass [10]. In Fig. 5(b) and (c), the benefits of adaptation are more visible as the class is made of a single image with complex frequency characteristics. Observe the good match between power spectrum of the image and the frequency response of the adapted filter. In Fig. 5(d), the class is made of 13 images with complex frequency characteristics.

The popularity of that class and others at node 9 is shown in Fig. 6(a). The figure also shows the frequency response of the adapted filter for each of the eight classes constructed by the classification algorithm. Similar plots are given in Fig. 6(b) for node 13, which features a smaller codebook (only two classes). The size of the codebook for all nodes of the tree is shown in Fig. 7. Observe that codebooks at nodes deep down the tree tend to be smaller. We conjecture that this is mainly due to the heavier penalty imposed by the cost function (9) on complex codebooks [as measured by the second entropy term in (9)] when the subband image size $N^{(n)}$ is small.

B. Adapted Tree and Quantizers

Node-split and quantizer codebooks were designed for five different values (0, 1, 10, 100, and ∞) of the Lagrange multiplier λ. The codebooks were designed as described in Section II-B. Each quantizer codebook initially contained 40 equally weighted, scalar uniform quantizers with step sizes in the set $\{5, 10, 15, \cdots, 200\}$. Note that this set is consistent with the dynamic range of the subband coefficients for typical classes of images of interest. The results of the quantizer codebook design are shown in Fig. 9. A significant benefit of the off-line training algorithm, in addition to the nonuniform codelengths of the quantizers based on their popularity of usage, is the reduction in on-line complexity that it affords by tailoring the set of quantizer choices at each node to the target quality level, as captured by the Lagrange multiplier λ. This is easily seen from Fig. 9, where the set of quantizers corresponding to the lower quality operating point $\lambda = 100$ is much coarser than the corresponding set associated with the higher quality operating point $\lambda = 10$. For each value of λ, a small subset of the initial large suite of quantizers is thus elegantly retained in an automatic manner. This subset is efficiently matched to the desired target bit rate without the need to resort to *ad hoc* methods based on empirical trial and error, as is typically done.

Node-split decisions were initially assumed to be equiprobable. The node-split codebooks on convergence of the classification algorithm are shown in Fig. 8 for all five values of λ.

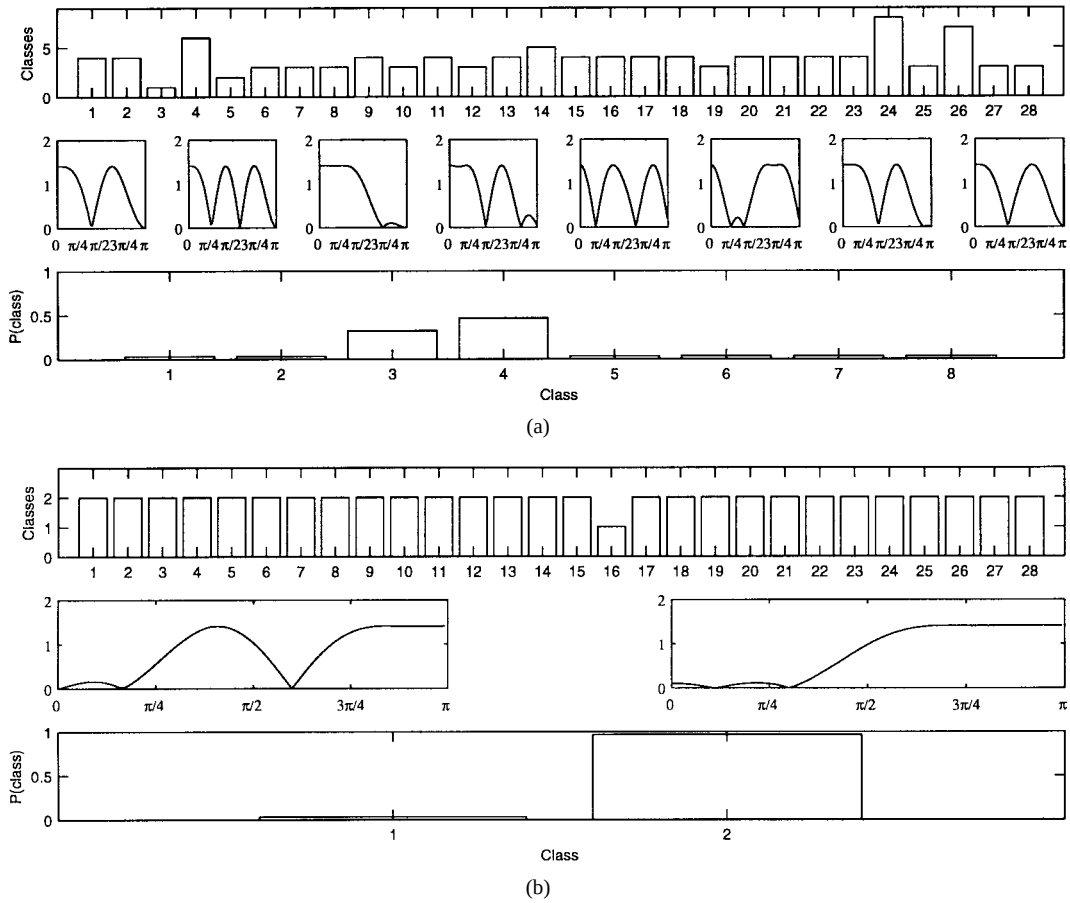


Fig. 6. "Vertical" filter bank codebooks for (a) node 9 (eight codewords) and (b) node 13 (two codewords). In each case, the top graph shows the class assignment for each training image. The middle and the bottom graphs, respectively, show the filter responses in the high-energy channel and the popularity of each class.

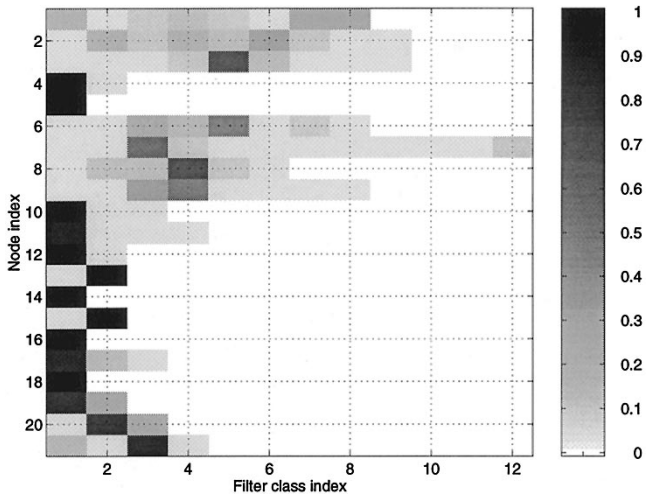


Fig. 7. "Vertical" filter bank codebook. Popularity of codewords at internal nodes of the subband tree. The popularity $p(i) \in [0, 1]$ is represented as a shade of gray; see intensity scale at right.

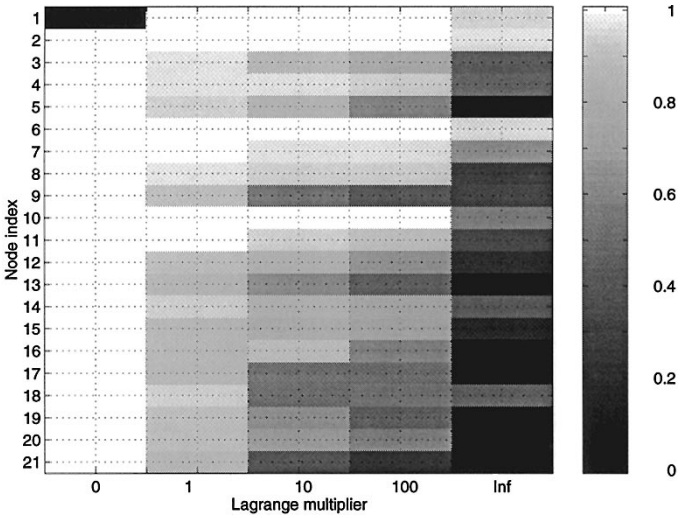
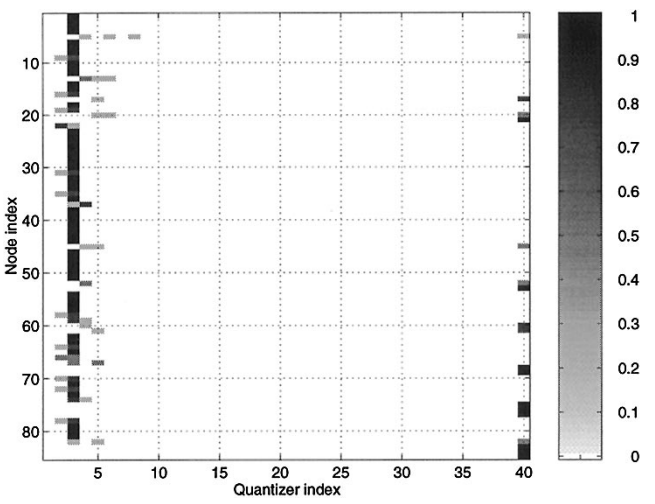


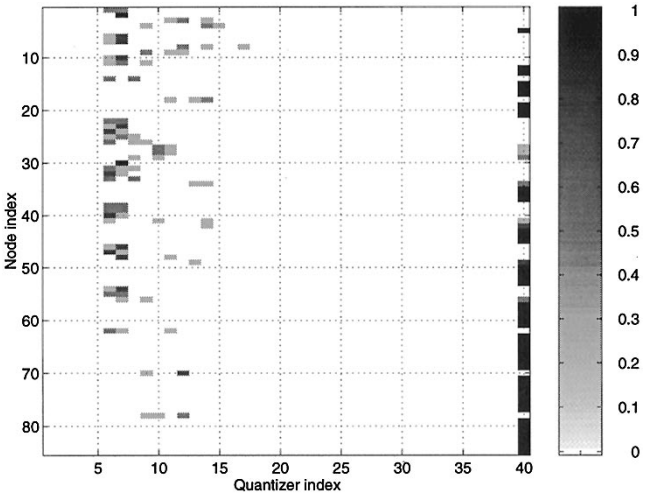
Fig. 8. Node-split codebook. The probability of each node split is represented as a shade of gray and is shown for five values of λ: 0, 1, 10, 100, and ∞.

As expected, there is a strong dependence on λ since node-split decisions become increasingly unlikely at low bit rates (high λ).

Storage Considerations: We would like to address the storage requirements associated with our algorithm. A definite bottleneck is the dependence of the codebook parameters on the rate-distortion slope "quality" criterion embodied by the Lagrange multiplier λ. Theoretically, we need a codebook for every value of λ. The storage burden associated with this is well-known from ECVQ codebook design theory (see the Appendix). One solution is to have only a few values of λ and deal with the issue of codebook mismatch (we discuss this in the next section). Another option is to build a tree-



(a)



(b)

Fig. 9. Quantizer codebooks for $\lambda = 10$ and $\lambda = 100$. Intensity represents quantizer popularity.

structured codebook that is analogous to the tree-structured vector quantizer and entropy-constrained tree structured vector quantizer codebook designs that are well known in the VQ community [13]. Modifications of our design methodology to a tree-structured framework remain an attractive avenue of future research to address the storage requirements for practical implementations. It is also useful to note that as the number of nodes increases exponentially with tree depth, using a different codebook at each node of a deep tree might present significant storage problems. Practical storage considerations might force us to limit tree depth to some reasonably small number, like 3 or 4.

C. Coding Results

To test our codebook design, we first selected a test image of one of the five types considered above: a fingerprint. The test image, shown in Fig. 10(a), was not part of the training set. We tested the performance of our on-line coder against that of the coder in [3] using Daubechies' nonadapted filter banks but adapted tree structures and quantizers [3]. We compared both coding schemes under bit rate constraints. We selected a target bit rate of 0.755 b/pixel, in which case, the Lagrange multiplier was $\lambda = 100$, and the appropriate codebook could be used. The coder from [3], which does not use a codebook, met the 0.755 b/pixel bit rate specification using $\lambda \approx 107$. The compressed images are shown in Fig. 11(a) and (b). We obtained a 0.54 dB PSNR improvement and substantial visual improvements as well. In particular, the ridges of the coded fingerprint in Fig. 11(a) are less jagged than those in Fig. 11(b). This result is consistent with earlier coding results using wavelet trees and fixed quantizers on textured images [6]. Similar results were obtained at other bit rates, e.g., using $\lambda = 10$, we obtained a bit rate of 1.828 b/pixel and a 0.42-dB improvement over [3]. The performance of our optimized coder was also compared with that of a coder using the same filter bank codebook but a simpler tree, which was obtained by recursively splitting the high-energy branch of the tree. At the target, bit rate of 0.755 b/pixel, we obtained a 0.70-dB improvement over the latter scheme. The results are summarized in Table I.

In another experiment, we investigated the potential effects of codebook mismatch. We encoded our test fingerprint image at higher rate ($\lambda = 50$), but since no codebook corresponding to that value of λ was designed, we successively used codebooks designed for $\lambda = 100$ and $\lambda = 10$ instead. In the first case, we obtained a bit rate of 0.900 b/pixel and a PSNR of 32.35 dB, which is a 0.45-dB improvement over the coder in [3] operating at the same bit rate (using $\lambda = 80$). In the second case, we obtained a bit rate of 1.392 b/pixel and a PSNR of 35.43 dB, which is a mere 0.07-dB improvement over the coder in [3] operating at the same bit rate (using $\lambda = 26$). This shows that there is a penalty for using a severely mismatched codebook. This issue is mitigated with the use of tree-structured codebooks, as discussed in the previous section.

We also applied the method to images such as *Dotted Array*[4] in Fig. 10(b) or *Lena* in Fig. 10(c), which are different from those in the training set. In that case, the coding results were comparable (slightly better or even slightly worse) with those using the coder in [3]. We believe this is due not only to a mismatch between the test image and the training set but to three other factors as well:

1) the use of the theoretical coding gain criterion for designing the filters, which does not guarantee improvements in actual R/D performance when standard assumptions are not satisfied;
2) the spatial inhomogeneity of some images (Section IV addressed some possible extensions to address this limitation);
3) the marginal gain of using adapted filter banks for some images. We discuss this important point in more detail below.

D. On Signal-Adapted Cascaded Filter Banks

Here, we describe some elementary properties of signal-adapted cascaded filter banks and compare them with traditional cascaded designs in terms of theoretical coding perfor-

[4]This image is from the MIT vision texture database, at http://www-white.media.mit.edu/vismod/.

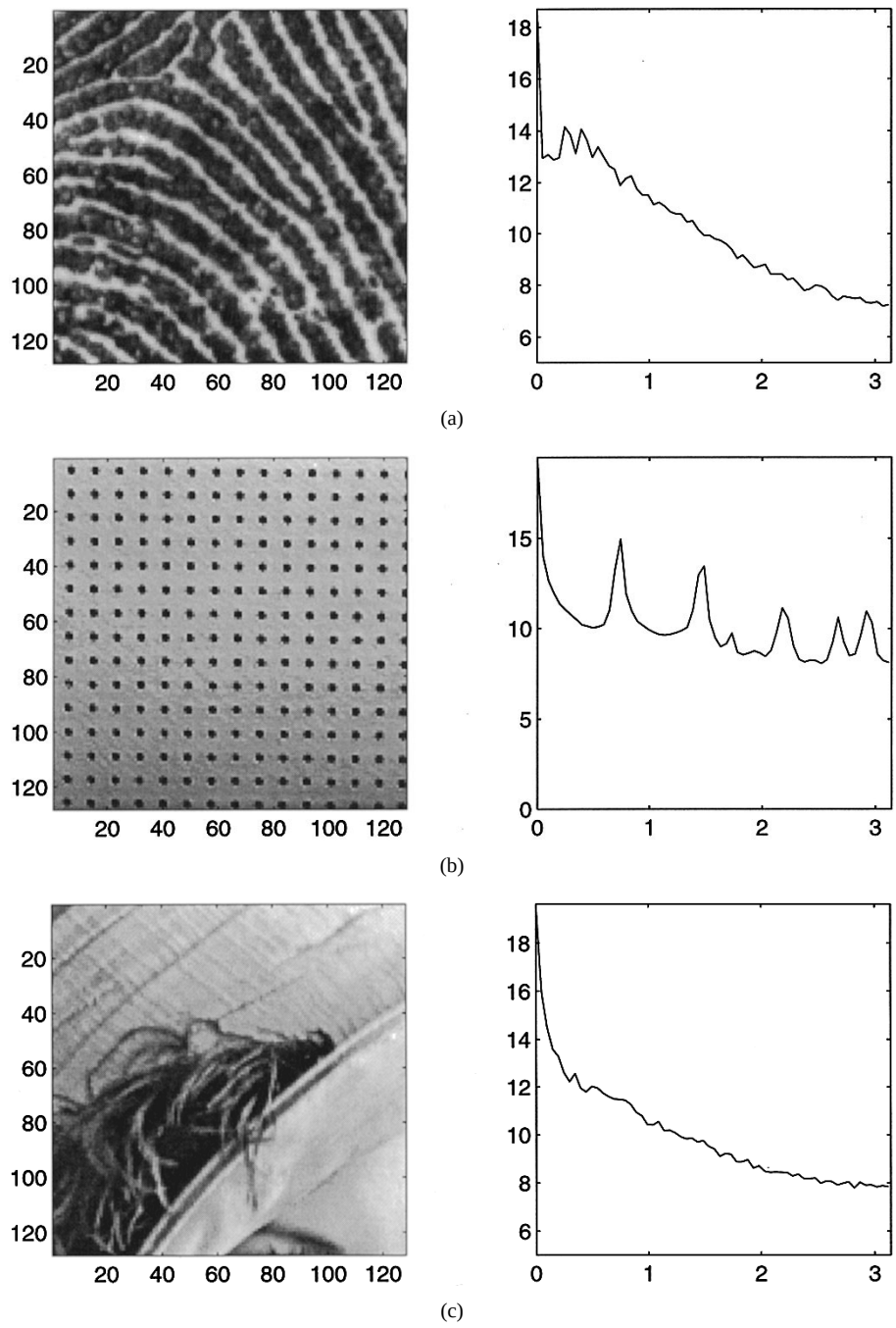


Fig. 10. Test images (128 × 128 pixels) and their horizontal log spectra. (a) Fingerprint. (b) Dotted array. (c) Lena.

mance. Consider, for simplicity, the case of unconstrained-length filters, which offer maximum theoretical performance, and 1-D signals with absolutely continuous spectral density $S(f)$. Assume that trees are limited to a fixed depth d. The performance of a hierarchical subband coder, measured by the log geometric mean $\mathcal{E} = (1/N)\sum_i N_i \ln \sigma_i^2$, approaches the R/D lower bound $\int \ln S(f)\, df$ when the spectrum of the input signal is relatively flat in each subband [17], [18]. If the standard subband coder already satisfies this flatness property, adaptive subband coding cannot provide tangible benefits. In contrast, if $S(f)$ has sharp variations *and* the tree depth d is small, the potential advantage of the adaptive scheme is significant. Below, we have some elementary properties of the log geometric means $\mathcal{E}_{ad}(d)$ and $\mathcal{E}_{cv}(d)$ for the adapted and conventional subband coding schemes, respectively. See [18] for typical values of $\mathcal{E}_{cv}(d)$ for AR(1) signals.

Property 1: $\int \ln S(f)\, df \le \mathcal{E}_{ad}(d) \le \mathcal{E}_{cv}(d)$ for all $d \ge 0$.

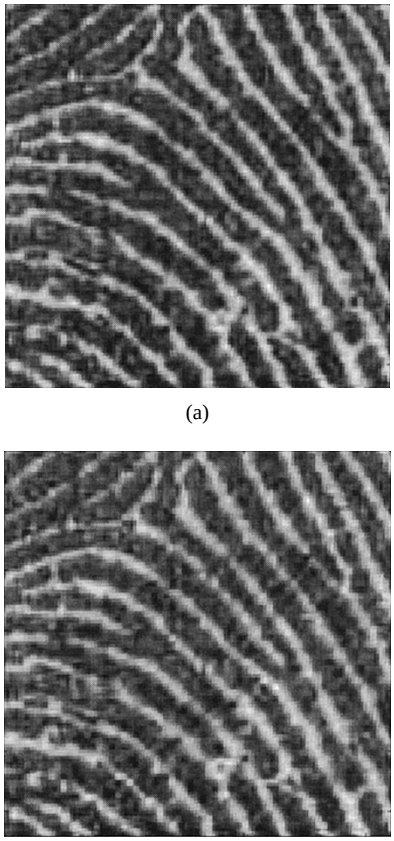
Property 2: The sequences $\mathcal{E}_{ad}(d)$ and $\mathcal{E}_{cv}(d)$ are nonincreasing in d.

Property 3: $\lim_{d\to\infty} \mathcal{E}_{ad}(d) = \lim_{d\to\infty} \mathcal{E}_{cv}(d) = \int \ln S(f)\, df$.

Note that the difference $\mathcal{E}_{cv}(d) - \mathcal{E}_{ad}(d)$ is not necessarily monotonic. Consider, for instance, $\ln S(f) = \mathcal{I}_{[0,\,1/16]\cup[2/16,\,3/16]}(|f|)$. In this case, $\mathcal{E}_{cv}(d) - \mathcal{E}_{ad}(d) \propto \delta(d-2)$. In addition, note that convergence of $\mathcal{E}_{ad}(d)$ and

TABLE I
CODING RESULTS USING OUR ADAPTIVE SUBBAND CODING METHOD (WITH $\lambda = 100$) AS WELL AS THE METHOD IN [3], USING DAUBECHIES' D4 FILTER BANKS

Image	Adapted tree				Wavelet tree			
	Adapted filters		D4 filters		Adapted filters		D4 filters	
	PSNR(dB)	R(bpp)	PSNR(dB)	R(bpp)	PSNR(dB)	R(bpp)	PSNR(dB)	R(bpp)
Fingerprint	31.29	0.755	30.75	0.775	30.72	0.753	30.02	0.743
Dotted array	29.26	1.126	29.04	1.126	29.28	1.205	29.45	1.195
Lena	31.48	0.555	31.61	0.553	31.70	0.636	31.32	0.610



(a)

(b)

Fig. 11. Coding fingerprint test image in Fig. 10(a) at 0.755 b/pixel. (a) Encoded using our adaptive subband coding method (PSNR = 31.29 dB). (b) Encoded using the method in [3], using Daubechies' D4 filter banks (PSNR = 30.75 dB).

$\mathcal{E}_{cv}(d)$ to the lower-bound limit is faster for smooth spectra. Hence, we conclude that adapted designs are unlikely to be worth the additional design complexity for deep trees or for relatively featureless spectra $S(f)$.

In our coding experiments, the tree depth $d = 3$. The log spectrum for *dotted array* in Fig. 10(b) exhibits distinctive features. Even though this test image is very different from those in the training set, the adaptive subband algorithm manages to identify filter banks that are well adapted to those spectral features, as indicated by the coding results in Table I. In contrast, not only is Lena very different from the images in the training set, but in addition, its log spectrum appears to be quite "featureless." The unimpressive numerical results for Lena are consistent with the observations above. Similar conclusions are expected to hold for images made of many different textured regions, as the distinctive spectral features of each region would be averaged out. In this case, the use of adaptive space-frequency trees could be advantageous due to their spatial adaptivity properties.

VI. CONCLUSION

We have explored the application of signal-adapted filter banks to image coding and proposed a new best-basis design in which filter banks, subband tree structure, and quantizers are chosen to optimize R/D performance. This design raises fundamental issues of library complexity, both in a coding sense (description of the chosen basis is potentially costly side information) and in a computational sense (efficient evaluation of all candidate bases). The problem is particularly involved because filter bank parameters belong to a continuum of admissible values. Some form of quantization of the filter bank parameters is necessary, but the construction of a suitable codebook of representative filter banks is itself a challenging problem. Our approach is based on training. We construct a codebook based on a set of training images using the theoretical coding gain approximation to R/D performance as the codebook design criterion. We also construct a codebook for the node-split decisions and quantizers at each node, using true R/D performance as the design criterion. Using the theoretical coding gain approximation and designing separate codebooks for filter banks and for the node-split decisions and quantizers is primarily motivated by practical considerations, owing to the near-intractability of the original R/D optimization problem. The encoder evaluates all candidate filter banks, node-split decisions, and quantizers and selects the best ones using a dynamic programming algorithm. There are typically, at most, a dozen candidate filter banks and quantizers per node.

Two salient computational advantages of the new approach (relative to [3]) are that 1) the on-line coding algorithm need evaluate only a suitable reduced set of candidate quantizers, and 2) our proposed framework allows for the potential of reducing the recursive evaluation of all possible trees to some optimal subtree (determined during the training phase). Since populating the full tree with Lagrangian costs is, in fact, the complexity bottleneck in [3], minimizing this phase is the key to reducing overall on-line complexity. Further complexity reduction may be obtained (at the cost of slight suboptimality) using simpler greedy top-down optimization algorithms [15].

We have obtained encouraging preliminary coding results on textured images such as fingerprints, both visually and in terms of PSNR (typically 0.5 dB improvement over [3]). However, in order to address the lack of spatial adaptivity of the current method (which we conjecture is one reason for unimpressive gains over [3] for spatially inhomogeneous

images), we suggest that the proposed paradigm should be extended to include spatial adaptivity. This involves the extension of the simple frequency trees in our current framework to more powerful space-frequency trees, as described in [5]. This approach requires the use of node-split decisions that indicate space or frequency splits. Optimal weights for these decisions may be determined by classification, extending the method described in Section II.

Another promising avenue of future research involves using an R/D rather than the theoretical coding gain criterion for filter bank design that, however, would lead to a considerable increase in both conceptual and computational complexity due to the complex interdependencies that would result. This could significantly enhance the performance of the theoretical framework presented here. As complexity of the algorithm is likely to increase extremely rapidly with tree depth, the use of limited-depth trees (for both theoretical and practical reasons) would improve the feasibility of such an approach. Short of achieving this ultimate goal, reliable optimization techniques should be developed to optimize the theoretical coding gain for cascaded filter banks. One of the limitations of the current algorithm is that the method for optimizing the global coding gain using "local" coding gain cost functions is suboptimal. While developing efficient optimization algorithms will ultimately establish the performance of the adaptive subband coding paradigm, the computational details of these optimization algorithms are somewhat orthogonal to the main contribution of our work: developing a systematic, R/D-based framework for selection and quantization of subband coder parameters, which provides a theoretically sound alternative to empirical designs.

Several extensions of our framework are possible. First, we would like to have the flexibility to choose from filters with different lengths. This cannot be done in the current framework because the theoretical coding gain criterion inevitably favors long filters. Second, using quantizers that are more sophisticated than the uniform scalar quantizers considered here would further improve coding performance, especially at low bit rates. Third, the current method requires that a different codebook be designed for every value of λ of interest, as in ECVQ [13]. A practical solution to that problem consists in designing tree-structured codebooks at the cost of a (possibly slight) suboptimality. The analogy is similar to unstructured entropy constrained vector quantization (ECVQ) versus tree-structured VQ (TSVQ) [13]. In the latter, a single tree-structured codebook is designed for all values of λ, with different pruned subtrees of the full-tree corresponding to different values of λ. Finally, codebook storage by the encoder and decoder might be a problem, especially if the tree is large; designing a size-constrained codebook from which filter banks at all nodes are selected would alleviate that difficulty.

APPENDIX A
ENTROPY-CONSTRAINED QUANTIZATION

Chou, Lookabaugh, and Gray proposed an iterative descent algorithm for designing vector quantizers having minimum distortion subject to an entropy constraint. Although we refer to [19] for a detailed description of this work, we briefly summarize here the steps of their iterative algorithm since it provides the basis for our approach to improving the performance of wavelet-based compression algorithms.

Let $\{\beta(i)\}_{i\in\mathcal{I}}$ denote an L-dimensional VQ codebook ($\beta(i) \in \mathcal{R}^L$) indexed by index set $\mathcal{I}$, and let $\{l(i)\}_{i\in\mathcal{I}}$ denote the transmitted symbol lengths of the codewords. Let a stationary vector source generate vectors $\{X^n\}$ from $\mathcal{R}^L$ independently with distribution P_{X^n}. Let λ be a Lagrange multiplier controlling the entropy constraint, and let $\rho(x, y)$ specify a distortion measure. Each iteration of the design algorithm consists of three steps, none of which can increase the value of the Lagrangian cost function. (Here, we ignore the processing associated with checking conditions for stopping the iteration).

1) Define an entropy-constrained mapping of input vectors (modified nearest neighbor condition)

$$\alpha(x) = \arg\min_{i\in\mathcal{I}} [\rho(x, \beta(i)) + \lambda l(i)].$$

2) Update transmitted symbol lengths to reflect codeword entropy (codelength update condition):

$$l(i) = -\log_2 P_{X^n}\{\alpha(X^n) = i\}.$$

3) Optimize reproduction codebook (centroid condition):

$$\beta(i) = E[X^n|\alpha(X^n) = i].$$

4) Iterate until stopped.

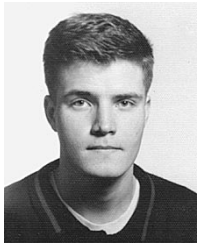
ACKNOWLEDGMENT

The authors would like to thank Prof. Mallat and an anonymous reviewer for helpful comments.

REFERENCES

[1] M. Vetterli and J. Kovačević, *Wavelets and Subband Coding.* Englewood Cliffs, NJ: Prentice-Hall, 1995.
[2] R. Coifman and V. Wickerhauser, "Entropy-based algorithms for best basis selection," *IEEE Trans. Inform. Theory,* vol. 38, pp. 713–718, Mar. 1992.
[3] K. Ramchandran and M. Vetterli, "Best wavelet packet bases in a rate-distortion sense," *IEEE Trans. Image Processing,* vol. 2, pp. 160–175, Apr. 1993.
[4] 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 Trans. Signal Processing, Special Issue on Wavelets,* vol. 41, pp. 3341–3359, Dec. 1993.
[5] Z. Xiong, K. Ramchandran, M. T. Orchard, and K. Asai, "Wavelet packets-based image coding using joint space-frequency quantization," in *Proc. ICIP,* Austin, TX, Nov. 1994, vol. 3, pp. 324–328.
[6] P. Delsarte, B. Macq, and D. T. M. Slock, "Signal-adapted multiresolution transform for image coding," *IEEE Trans. Inform. Theory,* vol. 38, pp. 897–904, Apr. 1992.
[7] H. Caglar, Y. Liu, and A. N. Akansu, "Statistically optimized PR-QMF design," *Proc. SPIE,* vol. 1605, pp. 86–94, 1991.
[8] M. Unser, "An extension of the Karhunen–Loève transform for wavelets and perfect-reconstruction filterbanks," *Proc. SPIE,* vol. 2034, pp. 45–56, 1993.
[9] P. Moulin, "A new look at signal-adapted QMF bank design," in *Proc. ICASSP,* Detroit, MI, 1995.
[10] P. Moulin, M. Anitescu, K. O. Kortanek, and F. Potra, "The role of linear semi-infinite programming in signal-adapted QMF bank design," *IEEE Trans. Signal Processing,* vol. 45, pp. 2160–2174, Sept. 1997.
[11] P. A. Chou, M. Effros, and R. M. Gray, "A vector quantization approach to universal noiseless coding and quantization," *IEEE Trans. Inform. Theory,* vol. 42, pp. 1109–1138, July 1996.

[12] M. Effros and P. A. Chou, "Weighted universal transform coding: Universal image compression with the Karhunen–Loève transform," in *Proc. ICIP*, 1995, pp. II.61–II.64.
[13] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression.* Boston, MA: Kluwer, 1992.
[14] I. Daubechies, *Ten Lectures on Wavelets.* Philadelphia, PA: SIAM, 1992, CBMS-NSF Regional Conf. Series Appl. Math., vol. 61.
[15] C. Taswell, "Satisficing search algorithms for selecting near-best bases in adaptive tree-structured wavelet transforms," *IEEE Trans. Signal Processing*, vol. 44, pp. 2423–2438, Oct. 1996.
[16] C. Herley, Z. Xiong, K. Ramchandran, and M. T. Orchard, "Joint space-frequency segmentation for least-cost image representation," *IEEE Trans. Image Processing*, vol. 6, pp. 1213–1230, Sept. 1997.
[17] N. S. Jayant and P. Noll, *Digital Coding of Waveforms.* Englewood Cliffs, NJ: Prentice-Hall, 1984.
[18] R. L. de Queiroz and H. S. Malvar, "On the asymptotic performance of hierarchical transforms," *IEEE Trans. Signal Processing*, vol. 40, pp. 2620–2622, Oct. 1992.
[19] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Entropy-constrained vector quantization," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 31–42, Jan. 1989.



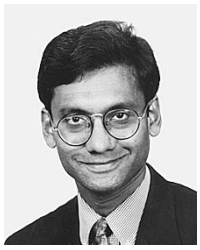
Vladimir Pavlovic (M'99) was born in Paris, France, in 1966. He received the Dipl. Eng. degree in electrical engineering from the University of Novi Sad, Yugoslavia, in 1991. In 1993, he received the M.S. degree in electrical engineering and computer science from the University of Illinois, Chicago. In 1999, he received the Ph.D. degree in electrical and computer engineering from the University of Illinois, Urbana-Champaign.

Since then, he has been with Compaq Computer Corporation (Cambridge Research Laboratory), Cambridge, MA. His research interests include Bayesian networks, time series modeling, and multimodal signal fusion.

Pierre Moulin (SM'98) received the D.Sc. degree from Washington University, St. Louis, MO, in 1990.

After working for five years as a Research Scientist for Bell Communications Research, Morristown, NJ, he joined the University of Illinois, Urbana-Champaign, where he is currently an Assistant Professor in the Department of Electrical and Computer Engineering and a Research Assistant Professor at the Coordinated Science Laboratory and the Beckman Institute's Image Formation and Processing Group. His fields of professional interest are image and video processing, statistical signal processing and modeling, and nonparametric function estimation; and the application of multiresolution signal analysis, optimization theory, and fast algorithms to these areas.

Dr. Moulin is an Associate Editor for the IEEE TRANSACTIONS ON INFORMATION THEORY in the area of nonparametric estimation, classification, and neural nets and a member of the IEEE Image and Multidimensional Signal Processing (IMDSP) Society Technical Committee. He received a 1997 CAREER Award from the National Science Foundation and the IEEE Signal Processing Society 1997 Best Paper Award in the IMDSP area.



Kannan Ramchandran (SM'98) received the B.S. degree from the City College of New York, New York, NY, and the M.S. and Ph.D. degrees from Columbia University, New York, NY, all in electrical engineering, in 1982, 1984, and 1993, respectively.

From 1984 to 1990, he was a Member of the Technical Staff at AT&T Bell Laboratories in the telecommunications and R&D area in optical fiber systems. From 1990 to 1993, he was a Graduate Research Assistant at the Center for Telecommunications Research, Columbia University. Since 1993, he has been with the University of Illinois, Urbana-Champaign, where he is currently an Assistant Professor in the Electrical and Computer Engineering Department and a Research Assistant Professor at the Beckman Institute and the Coordinated Science Laboratory. His research interests include image and video processing, multirate signal processing and wavelets, and signal processing-based approaches to communications and networking.

Dr. Ramchandran was the recipient of the 1993 Elaihu I. Jury Award at Columbia University for the best doctoral thesis in the area of systems, signal processing, or communications. He received an NSF Research Initiation Award in 1994, an Army Research Office Young Investigator Award in 1996, an NSF CAREER Award in 1997, and an ONR Young Invesigator Award in 1997. He received the 1996 Senior Best Paper Award from the IEEE Signal Processing Society for a paper with M. Vetterli and was selected as a Hank Magnuski Scholar for 1998 by the Department of Electrical Engineering at the University of Illinois. He is a member of the IEEE Image and Multidimensional Signal Processing (IMDSP) Committee and serves as an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING.