Hierarchical Quantization Indexing for Wavelet and Wavelet Packet Image Coding

Hasan F. Ates[*,1], Engin Tamer[1]

Dept. of Electronics Engineering, Isik University, Sile, 34980 Istanbul, TURKEY.

Abstract

In this paper, we introduce the quantization index hierarchy, which is used for efficient coding of quantized wavelet and wavelet packet coefficients. A hierarchical classification map is defined in each wavelet subband, which describes the quantized data through a series of index classes. Going from bottom to the top of the tree, neighboring coefficients are combined to form classes that represent some statistics of the quantization indices of these coefficients. Higher levels of the tree are constructed iteratively by repeating this class assignment to partition the coefficients into larger subsets. The class assignments are optimized using a rate-distortion cost analysis. The optimized tree is coded hierarchically from top to bottom by coding the class membership information at each level of the tree. Context-adaptive arithmetic coding is used to improve coding efficiency. The developed algorithm produces PSNR results that are better than the state-of-art wavelet-based and wavelet packet-based coders in literature.

Key words: image coding, wavelets, wavelet packets, classification, hierarchy.

1. Introduction

Within the last two decades, wavelet-based image coders have surpassed other transform-based coders in their coding efficiency. An image can be well-approximated by a sparse set of clustered significant coefficients in wavelet domain, and intelligent coding tools can be designed to reduce the bitrate required for coding this set. Among such tools, hierarchical zero-trees in EZW [1] and SFQ [2], set partitioning in hierarchical trees, i.e. SPIHT [3], and its modified versions [4, 5], and block partitioning of wavelet subbands in EZBC [6] are especially worth mentioning. The same partitioning tools could be easily adapted to be used in wavelet packets, resulting in coders such as WP&SFQ [7], WP-SPIHT [8, 7] and WP-BPO [9].

All of these successful wavelet and wavelet packet coders share a similar approach in how they handle the wavelet domain information during coding. The coefficients are partitioned/classified into significant and insignificant sets, and this partitioning information is embedded into the coded bitstream quite efficiently by the use of data structures

[*]Corresponding author. Tel.:+90-216 528 7133. Fax:+90-216 712 1472.
Email addresses: hfates@isikun.edu.tr (Hasan F. Ates), engin.tamer@isik.edu.tr (Engin Tamer)
[1]This research was supported by Isik University BAP-05B302 Grant.

such as zero-trees. As a result, large sets of insignificant (i.e. zero-quantized) coefficients are coded with little bitrate using this "partitioning map". The remaining much smaller number of coefficients are labeled as significant and coded using scalar quantization. In other coders [10], this partitioning is generalized to more than two classes of wavelet coefficients. However, it is important to minimize the additional coding cost of the classification map for any meaningful coding gain by using multiple classes.

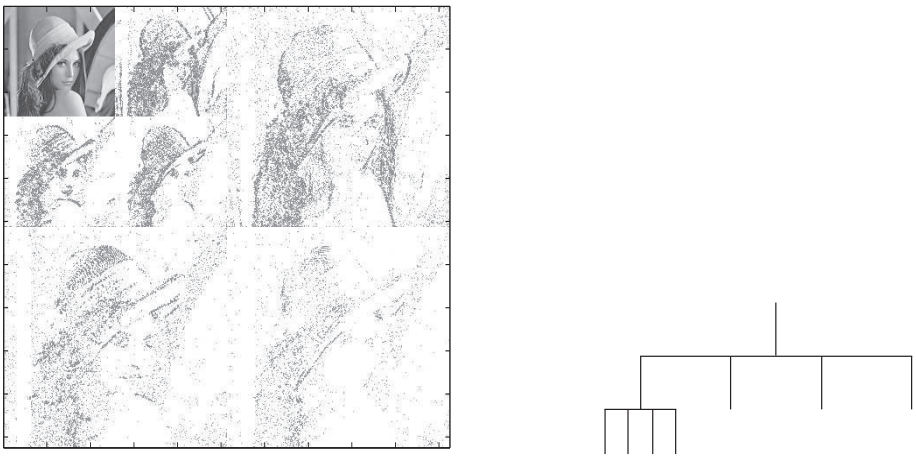
The side information required for any detailed classification of wavelet subbands becomes a bottleneck for coding especially at low bitrates. On the other hand, efficient bit allocation within the wavelet subbands requires an accurate characterization of the statistics of different regions in each subband. For instance, EQ coder [11, 12] makes use of local variance estimate to model the local statistics for more accurate bit allocation and superior coding efficiency. In spherical coding algorithm [13, 14], we show that a hierarchical refinement of the mean local energy achieves efficient bit allocation with no need for any side information.

In this paper, we extend our work in [15] and introduce both wavelet and wavelet packet coding algorithms that are based on hierarchical classification of wavelet coefficients using their quantization indices. After scalar quantization, wavelet coefficients are locally grouped together based on their quantization levels. A hierarchical classification map is defined in each wavelet subband, which describes the quantized data through a series of index classes. This hierarchical quantization index tree resembles the hierarchical energy tree of the spherical coder [14], except that it provides an exact knowledge of how each coefficient is quantized without any ambiguity. Going from bottom to the top of the tree, neighboring coefficients are combined to form classes that represent some statistics of the quantization indices of these coefficients. Higher levels of the tree are constructed iteratively by repeating this class assignment to partition the coefficients into larger subsets. At each level of the tree, the class assignment of a given subset describes some local statistics of the quantization levels of corresponding coefficients. This tree is coded hierarchically from top to bottom by coding the class membership information at each level of the tree.

The use of this quantization index hierarchy achieves accurate and efficient bit allocation within each subband without the need for any additional partitioning information. The bitrate required to code the class membership of a group of coefficients is proportional to the mean quantization level of these coefficients. Therefore, a higher amount of bitrate is spent for coding parts of the subband with higher number of significant coefficients, which leads to implicit adaptive bit allocation. For better bit allocation, the coding efficiency of the index tree is optimized based on a simple rate-distortion cost analysis.

Our coding algorithm uses context-adaptive arithmetic coding to adapt the coding of each class membership to the available local information. The class index and hierarchy level provide natural contexts for adaptive coding. As an improvement to [15], inter- and intra-band dependencies are also exploited by adapting the local coding model based on the spatial and scale neighborhood of each class assignment. In particular, we show that relative change in mean quantization levels are correlated within local neighborhoods, which can be used to improve the coding efficiency of the algorithm.

Section 2 provides brief background information about wavelet transform and wavelet packets. Section 3 in-



(a) (b)

Figure 1: For *Lena* image: (a) 2-level wavelet transform; (b) quadtree representation for the wavelet transform.

troduces the hierarchical classification concept and describes the hierarchical index tree. Then, Section 4 explains the details of the coding algorithm based on the quantization index hierarchy. Section 5 provides the details of the context-adaptive coding approach. In Section 6, the performance of the coding algorithm is evaluated by using two different classification strategies and in comparison to the coding efficiency of the state-of-art wavelet and wavelet packet coders.

2. Wavelet Transform and Wavelet Packets

Wavelet transform partitions an image into a hierarchical set of subbands, where each subband represents the frequency content of the image at a certain scale and orientation. One level of 2-D wavelet transform results in four subbands, i.e. low-pass subband and vertical, horizontal, diagonal detail subbands (see Figure 1). Based on multiresolution analysis, higher level transformations are obtained by iterative decomposition of the low-pass subband at each scale.

Wavelet packets are generalized form of wavelet transform and they are first presented by Coifman *et al.* in [16]. As mentioned above, in wavelet transform, quadrature mirror filter pairs are applied to low-pass subband and then decimation takes place in each scale. In wavelet packets, filter pair is applied to both low-pass and high-pass subbands and the results are decimated in all four subbands in every scale (see Figure 2(a)). If the subband decomposition is represented by a quadtree structure, wavelet transform splits one branch out of four in every node (see Figure 1(b)), while wavelet packets split all four branches making a full quadtree structure (see Figure 2(c)). Wavelet transform corresponds to just one possible pruned subtree of this full quadtree.

In wavelet packets, instead of using a fixed basis like in wavelet transform, a basis that is most suitable for the given image characteristics is chosen from the rich set of orthonormal bases. This selection process is performed by

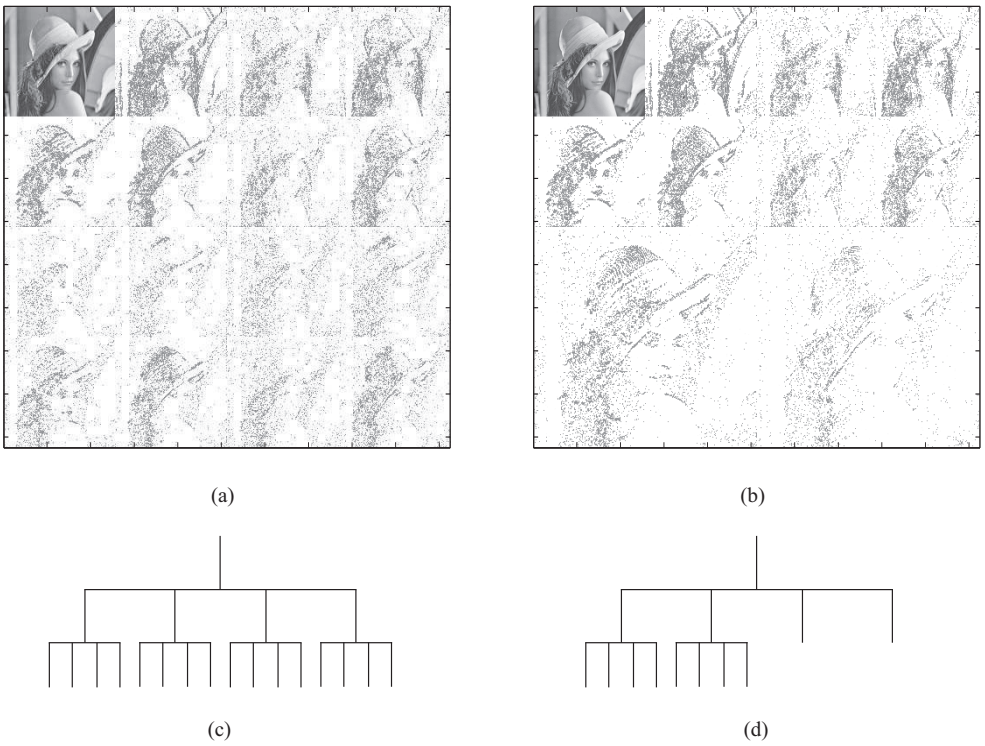


Figure 2: For *Lena* image: (a) 2-level full wavelet packet; (b) 2-level optimally pruned wavelet packet; (c) quadtree representation for the full wavelet packet; (d) quadtree representation for the pruned wavelet packet.

best basis selection algorithm. This algorithm can be seen as pruning of a full quadtree by using a certain cost criteria for obtaining the best representation of the image [7]. Several different cost functions, such as number of significant coefficients, entropy or logarithm of energy, etc., can be used as a decision criteria in pruning. For the best basis selection algorithm, we choose the logarithm of energy as the cost function:

$$C(W) = \sum_{m=1}^{2^J} \sum_{n=1}^{2^J} \log(c(m,n)^2) \tag{1}$$

where $c(m,n)$ represents the coefficients of a given wavelet subband W of size $2^J \times 2^J$. Figures 2(b),(d) show the subband decomposition for *Lena* based on this basis selection algorithm.

Wavelet packets provide more flexibility than wavelet transform for handling the space-frequency trade-off in regions where fine frequency localization is important, such as textured areas. In other words, wavelet packets exhibit good frequency localization at high frequencies as well as at low frequencies. Therefore, wavelet packets that are adapted to different frequency content of different images allow us to control the distribution of energy to coefficients better than wavelet transform. As a result, clustering of total energy in fewer coefficients increases the efficiency of the coding algorithm.

3. Hierarchical Quantization Index Classes

In this section we provide the notation for and analytical description of the index tree. As before, suppose that $c(m,n)$ represents the wavelet coefficients of the subband W in k-level wavelet or wavelet packet transform of a given image. W is of size $2^J \times 2^J$; then, $0 \leq m,n < 2^J$. The absolute value of each coefficient, i.e. $|c(m,n)|$, is scalar quantized and assigned to a non-negative quantization level/index $\Gamma(m,n)$, as follows:

$$\Gamma(m,n) = Q\left[|c(m,n)|\right], \tag{2}$$

$$\tilde{c}(m,n) = \text{sign}(c(m,n))Q^{-1}\left[\Gamma(m,n)\right]. \tag{3}$$

where $\tilde{c}(m,n)$ represents the reconstructed coefficient value by inverse quantization.

We would like to classify groups of wavelet coefficients according to their quantization indices. For a hierarchical representation, the classification starts with a pair of neighboring coefficients. Each such pair will be assigned to an index class, as defined below:

$$C_r = \{(i_1,i_2)|f(i_1,i_2) = r, \forall i_1,i_2,r \in \mathcal{Z}^+\} \tag{4}$$

where i_1, i_2, r are non-negative integers and $f(.,.)$ represents a class assignment function. As shown below, when this class definition is applied to the quantized subband, the integers i_1, i_2 represent the quantization indices of neighboring coefficients. The number of integer pairs in each class is given as $N_r = |C_r|$. The assignment function is supposed to classify similar index pairs under the same class. Section 3.2 explains how this "similarity" of index pairs could be defined.

These index classes will be used to construct a hierarchical description of the quantization indices of wavelet coefficients. Our goal here is to provide a coarse-to-fine refinement of local quantization information in subband W. We will show that this refinement achieves implicit bit allocation within the subband by assigning more bits to parts of the subband with higher number of significant coefficients. Note that, this hierarchical index tree is defined separately for each subband, and it is not related to the zero-trees used in EZW [1] to represent the parent-children hierarchy between coefficients at different scales.

The first level of the hierarchical index tree is formed by pairing neighboring wavelet coefficients according to the class definition given above:

$$\begin{aligned} \Gamma_{0,0}(m,n) &= Q\left[|c(m,n)|\right], \\ \Gamma_{1,0}(s,t) &= f(\Gamma_{0,0}(2s,t), \Gamma_{0,0}(2s+1,t)). \end{aligned} \tag{5}$$

Note that, to avoid any confusion between spatial coordinates of index tree and subband coordinates, (s,t) is used instead of (m,n) in the definition of the index hierarchy. Likewise, upper levels of the hierarchy are defined iteratively as follows (for $0 < u \leq J$):

$$\begin{aligned} \Gamma_{u,u}(s,t) &= f(\Gamma_{u,u-1}(s,2t), \Gamma_{u,u-1}(s,2t+1)), \\ \Gamma_{u+1,u}(s,t) &= f(\Gamma_{u,u}(2s,t), \Gamma_{u,u}(2s+1,t)). \end{aligned} \tag{6}$$

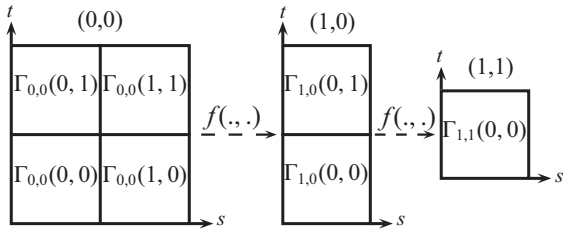


Figure 3: Hierarchical index tree ($J = 1$).

In this paper, we propose to code this hierarchical index tree, instead of coding the quantization index of each wavelet coefficient individually. More specifically, from top to bottom, the class assignment values, $\Gamma_{u,v}(s,t)$ $(0 \le u \le J, v \in \{u, u-1\}, 0 \le s,t < 2^{J-u}, 2^{J-v})$, will be coded. Figure 3 shows the class assignment variables at different levels of the index hierarchy for $J = 1$. Starting with $\Gamma_{1,1}(0,0)$, the class assignments from (6) are used to decode in turn $\Gamma_{1,0}$ and $\Gamma_{0,0}$, which is equal to the quantization indices of the wavelet subband. The details of the coding procedure will be explained in Section 4.

3.1. Quantization Function

The quantization function $Q[.]$ could be selected as any scalar quantizer. In this paper, we prefer to use a dead-zone uniform quantizer, since the use of a dead-zone improves the coding efficiency by fine tuning the set of insignificant wavelet coefficients that are to be quantized to zero. Hence:

$$Q[c] = \begin{cases} 0 & \text{if } 0 \le c < T \\ \left\lfloor \frac{c-T}{q} + 1 \right\rfloor & \text{if } T \le c \end{cases} \tag{7}$$

where T is the deadzone size and q is the quantization step size. The inverse quantization is given as:

$$Q^{-1}[i] = \begin{cases} 0 & \text{if } i = 0 \\ iq + T - q/2 & \text{else} \end{cases} \tag{8}$$

3.2. Class Assignment Function

In classification of wavelet coefficients, the main goal is to differentiate coefficients based on their statistical properties or information content. Therefore, the class assignment is supposed to represent some common statistics of the wavelet coefficients that are assigned to the same class. In other words, groups of coefficients with similar information content should be assigned to the same class. Depending on how this information content is described, many class assignment functions can be designed. In this paper, we look at two different assignment functions that provide high coding efficiency.

The first function is motivated by the local energy representation in spherical coder, and assigns index pairs (i_1, i_2) to the closest circle with integer radius (see Figure 4(a)):

$$f(i_1, i_2) = \left\lfloor \sqrt{i_1^2 + i_2^2} + 0.5 \right\rfloor \tag{9}$$

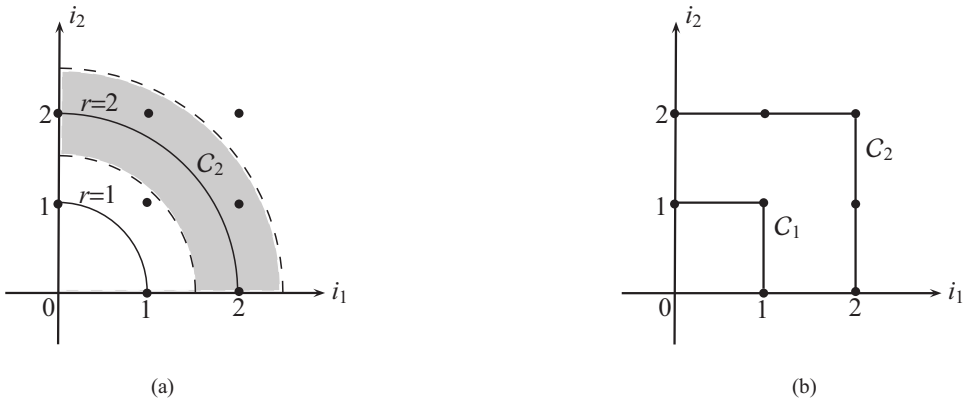


Figure 4: (a) Circular assignment classes; (b) Max assignment classes.

Hence, at each level of the hierarchy, $\Gamma_{u,v}(s,t)/2^{u+v}$ will approximately represent the root mean square of the quantization indices of the underlying coefficients. In Figure 4(a), the gray area shows the class C_2 and the dots in this area show the corresponding index pairs (i.e. $(0,2)$, $(2,0)$, $(2,1)$, $(1,2)$ and $N_2 = 4$) for this "circular" assignment function.

Another successful choice for $f(.,.)$ is the maximum of the two indices:

$$f(i_1, i_2) = \max(i_1, i_2) \tag{10}$$

This time $\Gamma_{u,v}(s,t)$ will correspond to the maximum index of the underlying coefficients. Figure 4(b) plots the set of solutions for $\max(i_1, i_2) = r$, when $i_1, i_2 \geq 0$ and $r = 1, 2$. There are $N_r = 2r+1$ integer pairs on these reverse L-shaped curves, which form the index classes C_r for this "max" assignment function. We will compare the performance of these two assignment functions in Section 6.

For both assignment functions, the pairs (i_1, i_2) for a class C_r (i.e. $f(i_1, i_2) = r$) could be indexed from 0 to $N_r - 1$ in the order of increasing angle; that is,

$$S_{(i_1,i_2)} > S_{(j_1,j_2)} \quad \text{if} \quad \arctan(i_1, i_2) > \arctan(j_1, j_2), \tag{11}$$

where $0 \leq S_{(i_1,i_2)}, S_{(j_1,j_2)} < N_r$, $0 \leq \arctan(i_1, i_2), \arctan(j_1, j_2) \leq \pi/2$. Figure 5 shows this indexing for the max assignment function, when $r = 2$. Note that, $S_{(r,0)} = 0$ and $S_{(0,r)} = N_r - 1$. This notation will be particularly useful in Section 5 when explaining the context-adaptive coding of the class assignment variables.

Note that, for both assignment functions, $f(0,0) = 0$ and $N_0 = 1$. These two conditions are essential for efficient class assignment, due to reasons that will be clarified in the next section. Section 4 describes a simple optimization and coding strategy for efficient coding of the quantization index hierarchy.

4. Subband Coding by Quantization Index Hierarchy

The coding algorithm described in this section is applied in each wavelet subband, W, of size $2^J \times 2^J$. The coded data are the class assignment variables, $\Gamma_{u,v}(s,t)$, plus the sign bits of each significant wavelet coefficient.

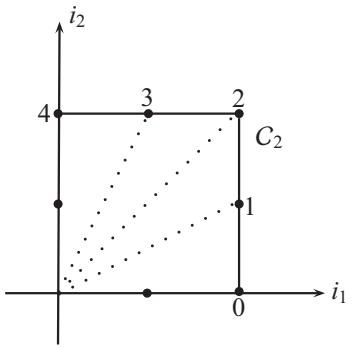


Figure 5: Indexing of class elements for max assignment ($r = 2$).

Encoding/decoding is performed hierarchically, starting from the top of the index tree, i.e. $\Gamma_{J,J}(0,0)$, going down to the coefficient level, i.e. $\Gamma_{0,0}(m,n)$.

For the vertical/horizontal subbands of the wavelet transform, coefficients are more correlated along the low-pass filtered direction. As a result, coding performance is slightly better when the first level of the index tree is built by pairing two neighboring indices along the low-pass direction. Therefore, to comply with the definition of index tree as given in Section 3, vertical subbands are transposed before the tree is built.

During encoding, at level (u,u) of the index hierarchy, given $\Gamma_{u,u}(s,t)$, we know from (6) that $(\Gamma_{u,u-1}(s,2t), \Gamma_{u,u-1}(s,2t+1))$ should be one of the $N_{\Gamma_{u,u}(s,t)}$ index pairs in class $C_{\Gamma_{u,u}(s,t)}$. Assuming all the index pairs are equally probable, entropy coding this class assignment information requires $\log_2(N_{\Gamma_{u,u}(s,t)})$ bits on average.

Note that, whenever a subtree of the index hierarchy is assigned to zero-class C_0 (i.e. $\Gamma_{u,v}(s,t) = 0$), all the subsequent class assignments and hence all the wavelet coefficients belonging to this subtree should be zero, and no additional bitrate is needed to code the remaining class indices of that subtree. Finding which subtrees should be assigned to C_0, i.e. pruning the index tree, is essential for improving the coding efficiency of the proposed algorithm.

Indeed, it turns out that building the hierarchical index tree using original indices and coding this tree does not lead to an optimal coding result. For instance, a coefficient could be individually considered as significant and quantized to a nonzero level. However, when considered as part of a subtree in which each nonzero class assignment will cost additional bitrate, it might actually reduce the total rate-distortion cost to zero-quantize this coefficient and all the other coefficients of this subtree. Hence, for optimal performance, the total coding cost of each subtree should be evaluated and it should be determined whether to code it as it is or assign all class indices to zero. However, the cost of upper level class assignments are dependent on the cost of lower level assignments, making optimal "pruning" of the index tree a difficult problem.

Since it is not computationally feasible to evaluate the cost of all possible prunings of the index tree for the globally optimal solution, we propose to use a greedy algorithm that performs a rate-distortion cost analysis similar to the one described in [14]. Going from the bottom to the top of the index tree, we compare the Lagrangian cost of zero-quantizing all coefficients of a given subtree to the best alternative associated with choosing not to do so. The

latter is equal to the cost of coding the class assignment of the current subtree plus the minimum costs of the two children subtrees. At the end, coefficients that belong to zero-classes are set to zero.

In more detail, the algorithm is given as follows (assume $0 \le m, n < 2^J$):

1. Quantize wavelet coefficients using a dead-zone quantizer:

$$\begin{aligned} \Gamma_{0,0}(m,n) &= Q\left[|c(m,n)|\right] \\ \tilde{c}(m,n) &= \text{sign}(c(m,n))Q^{-1}\left[\Gamma_{0,0}(m,n)\right] \end{aligned}$$

2. **Optimizing index tree:** For each subtree, we compare the Lagrangian cost of sending class assignments of wavelet coefficients to the cost of quantizing them all to zero. If the latter cost is smaller, then we assign that subtree to C_0. Suppose $L_{u,v}(s,t)$ represents the Lagrangian cost. At the bottom of the tree, given $\Gamma_{0,0}(m,n)$, the only bitrate required is for coding the signs of nonzero wavelet coefficients. Then,

$$L_{0,0}(m,n) = (c(m,n) - \tilde{c}(m,n))^2 + \lambda I(m,n)$$

where $I(m,n)$ represents the sign bit cost for coefficient $c(m,n)$, i.e.

$$I(m,n) = \begin{cases} 0 & \text{if } \Gamma_{0,0}(m,n) = 0 \\ 1 & \text{else} \end{cases}$$

As the rate-distortion cost analysis progresses from the bottom to the higher levels of the hierarchy, the additional bitrate required at each level (u,v) is equal to the sum of class assignment costs, i.e. $\log_2(N_{\Gamma_{u,v}(s,t)})$. Therefore, for each subtree at level (u,v), the total Lagrangian cost, $L_{u,v}(s,t)$, is equal to $\lambda \log_2(N_{\Gamma_{u,v}(s,t)})$ plus the total costs of its two children subtrees at the previous level. Based on this observation, the greedy optimization continues as follows:

Set $u = 1$. While $u < J$ do,

- For $0 \le s < 2^{(J-u)}$, $0 \le t < 2^{(J-u+1)}$, define $\Gamma_{u,u-1}(s,t)$ according to (6), and define the Lagrangian costs:

$$L_{u,u-1}(s,t) = L_{u-1,u-1}(2s,t) + L_{u-1,u-1}(2s+1,t) + \lambda \log_2(N_{\Gamma_{u,u-1}(s,t)})$$

  where the Lagrangian cost for coding class assignment, $\lambda \log_2(N_{\Gamma_{u,u-1}(s,t)})$, is added to the total cost of two children subtrees in order to get the total cost of current subtree. Then, we compare this cost to the total distortion caused by zero-quantization, and if the latter is smaller the current subtree is pruned:

$$L_{u,u-1}(s,t) > \sum_{m=2^u s}^{2^u(s+1)-1} \sum_{n=2^{u-1}t}^{2^{u-1}(t+1)-1} c(m,n)^2 \quad \Rightarrow \quad \Gamma_{u,u-1}(s,t) = 0.$$

- For $0 \le s, t < 2^{(J-u)}$, repeat the same procedure for $\Gamma_{u,u}(s,t)$ and $L_{u,u}(s,t)$.
- Increment u and repeat.

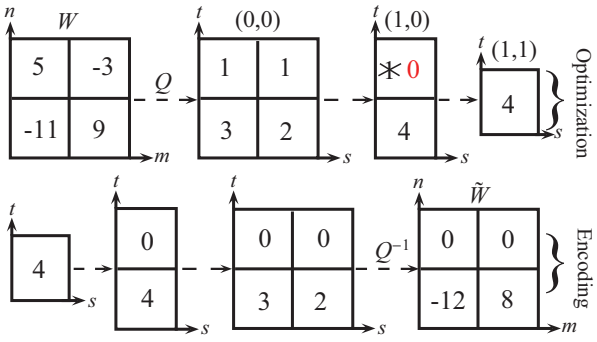


Figure 6: Optimizing and coding the index tree ($J = 1$).

3. **Encoding/Decoding:** Code $\Gamma_{J,J}(0,0)$. Set $u = J$. While $u > 0$ do,
   - For $0 \le s, t < 2^{(J-u)}$, encode/decode the subtree assignments, $\Gamma_{u,u-1}(s, 2t)$ and $\Gamma_{u,u-1}(s, 2t+1)$.
   - For $0 \le s < 2^{(J-u)}$, $0 \le t < 2^{(J-u+1)}$, encode/decode the subtree assignments, $\Gamma_{u-1,u-1}(2s, t)$ and $\Gamma_{u-1,u-1}(2s+1, t)$.
   - Decrement u and repeat.
4. Code the sign information if $\Gamma_{0,0}(m,n) > 0$. At the end of encoding/decoding, we reconstruct the decoded wavelet coefficients:

$$\tilde{c}(m,n) = \text{sign}(c(m,n))Q^{-1}\left[\Gamma_{0,0}(m,n)\right]$$

In the algorithm, q and T are chosen as the optimal quantization step size and the optimal dead-zone interval size, respectively, for best rate-distortion performance for a given Lagrangian multiplier λ. For a given bitrate, optimal λ is found using the convex bisection algorithm of [17].

While decoding the final index tree, once the algorithm reaches to a subtree in "zero-class" ($\Gamma_{u,v}(s,t) = 0$), all the coefficients that belong to that subtree are set to zero and no further bitrate is spent for coding the remaining class indices of the subtree. Therefore, the Lagrangian cost analysis to determine the subtrees in "zero-class" is essential for achieving successful coding results.

Figure 6 shows a 2×2 toy example for the optimization and coding steps of the algorithm, in which $q = 4$, $T = 2$, $\lambda = 10$ and circular assignment is used. For $\Gamma_{0,0}(0,1) = 1$ and $\Gamma_{0,0}(1,1) = 1$, we get $\Gamma_{1,0}(0,1) = f(1,1) = 1$. However, since $L_{0,0}(0,1) = (5-4)^2 + 10$, $L_{0,0}(1,1) = (-3+4)^2 + 10$, and $N_1 = 3$, the total cost for this assignment becomes $L_{1,0}(0,1) = 22 + 10\log_2(3) = 37.85$, which is larger than the cost of "zero-class", i.e. $5^2 + 3^2 = 34$. Therefore, we set $\Gamma_{1,0}(0,1) = 0$ and the corresponding coefficients are quantized to zero.

Note that, this greedy algorithm makes locally optimal pruning decisions as the subtrees are evaluated from the bottom to the top levels of the hierarchy. If a locally optimal decision is not globally optimal, then it must be true that a locally higher cost decision leads to a lower overall cost. Since rate-distortion costs are accumulated between consecutive hierarchy levels, this is only possible when a nonzero class assignment is switched to zero-class, which might reduce the assignment costs of upper levels. Hence "preemptive pruning" at the lower levels could in theory

avoid excessive pruning at the upper levels. However, as long as q and T are optimally chosen for a given λ, this is a marginal situation, and the greedy algorithm converges to a near-optimal solution.

Even though the hierarchical structure and the rate-distortion optimization of the index tree coder resembles our spherical coder in [14], there are also some fundamental differences between the two coding approaches. First of all, the spherical coding tree is built upon the original wavelet coefficients, and the coded variables are the actual local sum of coefficient energies, instead of being some function of local quantization indices as in the index tree. As a result, the spherical tree coding is a lossy coding approach in which the final coded values are not directly related to the scalar quantized values of wavelet coefficients. Even though this approach has its merits as shown by the theoretical analysis in [14], the nonlinear coding structure of the spherical tree makes it hard to fine tune the actual coded values of the wavelet coefficients. For instance, during the encoding/decoding stage, many wavelet coefficients end up being quantized to zero, even though rate-distortion optimization stage doesn't enforce them to be zero. This leads to a suboptimal coding outcome where coefficients are zero-quantized more than they should be. On the other hand, the index tree coder codes losslessly the quantization indices after the rate-distortion optimization step; hence there is no ambiguity about the outcome of the index-tree coding approach.

Another advantage of the index tree coder is the discrete nature of the tree nodes, i.e. class assignment indices. While the spherical tree nodes are continuous-valued, index tree nodes can take on a finite number of values as determined by the assignment classes. This not only makes lossless coding of the indices possible but also provides a flexible framework where many different node assignments can be investigated. In this paper, we have examined two such approaches, namely the circular assignment and max assignment functions.

Next section introduces another major contribution of the index tree coding approach. Since coded coefficient values can be determined without any ambiguity during the construction and encoding of the index tree, it becomes much easier to develop new coding tools that can exploit intra-band dependencies among wavelet coefficients for better coding efficiency.

5. Context-adaptive Arithmetic Coding

Context-adaptive arithmetic coding is used to code the class assignment variables $\Gamma_{u,v}(s,t)$. The index tree provides a natural context for adaptive arithmetic coding. First of all, the coding model of each class $C_{\Gamma_{u,v}(s,t)}$ depends on the corresponding number of index pairs, $N_{\Gamma_{u,v}(s,t)}$. This model adaptation is one of the main reasons behind the success of implicit bit allocation in hierarchical index tree coding.

Next, the coding model of $C_{\Gamma_{u,v}(s,t)}$ is also adapted based on the level of the tree, i.e. $lev = u + v$, since histograms of assignment classes are slightly different at different levels of index tree. Figure 7 shows the histogram of indices $S_{(i_1,i_2)}$, as defined in Section 3.2, for $lev = 2$ and $lev = 5$ in *Lena* image, when $r = 1$ and circular assignment is used. As seen from this example, the histogram gets more peaked around $S_{(i_1,i_2)} \approx N_r/2$ (i.e. $\arctan(i_1, i_2) = \pi/4$) at higher levels of the hierarchy. Model adaptation based on the tree level helps arithmetic coder adapt to the appropriate

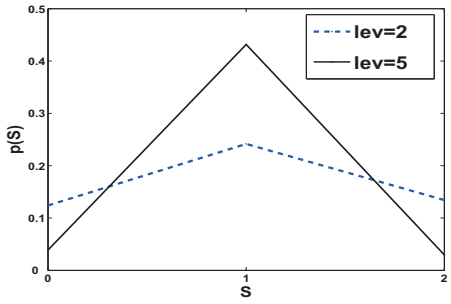


Figure 7: Histograms of class indices for $lev = 2$ and $lev = 5$ ($r = 1$, *Lena* at 1 bpp).

histogram and improve coding efficiency. Since histograms begin to look similar and there are much fewer number of variables at higher levels of the tree, this level adaptive modeling is used only for lower levels, i.e. when $0 \leq lev \leq 7$.

The causal spatial and scale neighbors of $\Gamma_{u,v}(s,t)$ also provide valuable contexts for adaptive coding. Since significant wavelet coefficients are clustered in space and in scale, there exists some correlation in the local distribution of wavelet coefficients at different scales. This leads to complicated dependencies among neighboring class assignments. In order to exploit these dependencies and achieve coding gain, we propose inter-band and intra-band conditional coding techniques, which are detailed below.

For exploiting the parent-children dependencies among two subbands at different scales, we note that $\Gamma_{u,v}(s,t)$ and $\Gamma^c_{u-1,v-1}(s,t)$ correspond to the same spatial region in the image, where c stands for the coarser scale subband and $lev = u + v \geq 2$. Therefore, the relative magnitudes of these two class assignments with respect to the neighboring regions are correlated. We exploit this using a simple contextual adaptation. We compare the two parent class indices; we adapt the coding model depending on which of the two indices is larger. Specifically, for $C_{\Gamma_{u+1,u}(s,t)}$, when $u \geq 1$ and $\Gamma^c_{u,u-1}(s,t) > 0$, the coded index S is chosen as follows:

$$S = \begin{cases} S_{(\Gamma_{u,u}(2s,t),\Gamma_{u,u}(2s+1,t))} & \text{if } \Gamma^c_{u-1,u-1}(2s,t) \geq \Gamma^c_{u-1,u-1}(2s+1,t) \\ S_{(\Gamma_{u,u}(2s+1,t),\Gamma_{u,u}(2s,t))} & \text{else} \end{cases} \tag{12}$$

When the relative magnitudes are correctly estimated from the parent class indices, we have $0 \leq S < N_r/2$. Hence the conditional histogram gets more peaked around these indices, which leads to lower entropy and, therefore, coding gain. When $\Gamma^c_{u,u-1}(s,t) = 0$, there is no available coarser scale information, and the standard coding model is used. A similar formulation is also used for $C_{\Gamma_{u+1,u+1}(s,t)}$ and its parent node. Figure 8 highlights the difference between the conditional histogram and the original histogram, when $r = 3$ and $lev = 5$. The difference in entropy between the two distributions is 0.1 bits.

For $lev = 0, 1$, there is no corresponding coarser scale parent and intra-band dependencies should be investigated. For $lev = 1$, the relative magnitudes of immediate neighboring class indices will be correlated. This can be seen through a simple example. In the horizontal subband, around a horizontal edge, the relative magnitudes of the coefficients to the above and to the below of the edge should be comparable throughout the full length of the edge. At

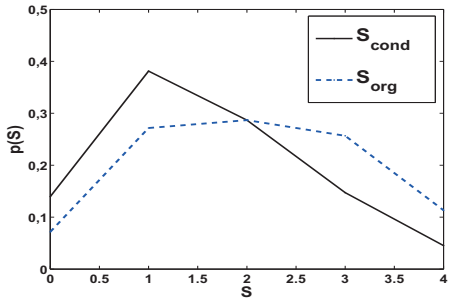


Figure 8: Conditional and original histograms for $r = 3$ and $lev = 5$ (*Lena* at 1 bpp).

$lev = 1$, we look at these horizontal neighboring class assignments, and once again choose conditional coding indices based on the relative magnitudes. In other words, for $C_{\Gamma_{1,1}(s,t)}$, and when $\Gamma_{1,1}(s-1,t) > 0$, the coded index is chosen as:

$$S = \begin{cases} S_{(\Gamma_{1,0}(s,2t),\Gamma_{1,0}(s,2t+1))} & \text{if } \Gamma_{1,0}(s-1,2t) \geq \Gamma_{1,0}(s-1,2t+1) \\ S_{(\Gamma_{1,0}(s,2t+1),\Gamma_{1,0}(s,2t))} & \text{else} \end{cases} \tag{13}$$

For $lev = 0$, there are complicated dependencies not only in relative magnitudes but also in signs of neighboring wavelet coefficients. Therefore, we modify the class assignments at $lev = 0$, in order to incorporate sign information into the definition of index tree, as follows:

$$\Gamma_{0,0}(m,n) = \text{sign}(c(m,n))Q\left[|c(m,n)|\right]. \tag{14}$$

Hence, for $lev = 0$, the assignment classes should be modified to account for negative quantization indices; we call these new classes as C_r^0:

$$C_r^0 = \{(i_1,i_2)|f(i_1,i_2) = r, \forall i_1,i_2 \in \mathcal{Z}, r \in \mathcal{Z}^+\} \tag{15}$$

where i_1, i_2 could be any integer. With this new definition, circular assignment function is still valid and the max function is changed to account for negative values, i.e. $f(i_1,i_2) = \max(|i_1|,|i_2|)$. The overall effect is that, the assignment classes now cover all of the four quadrants, instead of being limited to the first quadrant as in Figures 4(a) and (b). Note that, $N_r^0 = 4N_r - 4$, since the index pairs $(r,0)$, $(0,r)$, $(-r,0)$, $(0,-r)$ should only be accounted for once. Also, we now have $0 \leq \arctan(i_1,i_2) < 2\pi$. With this modification, sign coding for significant wavelet coefficients also become a part of the index tree coding, particularly the coding of class indices for $C^0_{\Gamma_{1,0}(s,t)}$.

With this modified definition, the complicated intra-band dependencies among wavelet coefficients could be better investigated. Specifically, we propose to estimate the angle of current class index, $\arctan(\Gamma_{0,0}(2s,t),\Gamma_{0,0}(2s+1,t))$, from the angle of its causal neighbor, i.e. $\arctan(\Gamma_{0,0}(2s,t-1),\Gamma_{0,0}(2s+1,t-1))$. Then, estimation error could be coded instead of the original class assignment. Overall, this operation could be seen as rotating the index pair $(\Gamma_{0,0}(2s,t),\Gamma_{0,0}(2s+1,t))$ by the negative of the estimated angle. However, to achieve perfect recovery of the original indices after decoding, we need to make sure that the index pair assigned after rotation is also a member of the class $C^0_{\Gamma_{1,0}(s,t)}$, and this pair could be rotated back to its original position with the decoded neighbors' angular information.

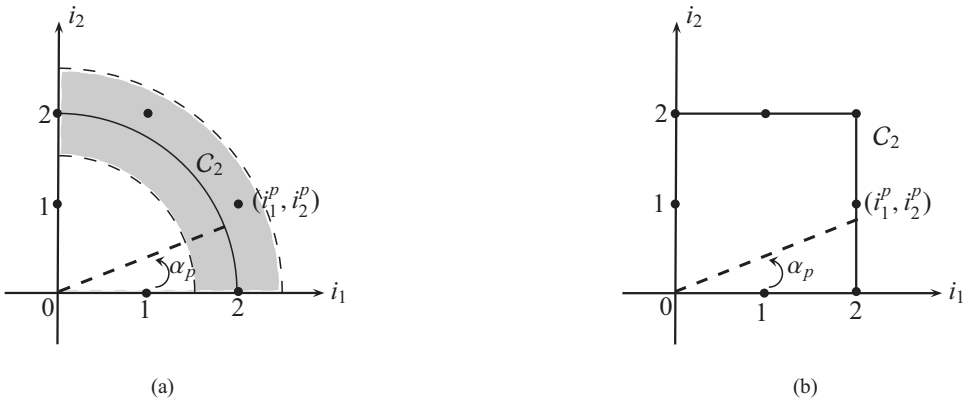


Figure 9: Angular prediction for: (a) Circular assignment; (b) Max assignment.

To satisfy these requirements, we propose the following conditional coding strategy; suppose $r = \Gamma_{1,0}(s,t)$, $i_1 = \Gamma_{0,0}(2s,t)$, $i_2 = \Gamma_{0,0}(2s+1,t)$, and α_p is the estimated angle:

$$\begin{aligned} (i_1^p, i_2^p) &= g(r, \alpha_p), \\ S_e &= \text{mod}\,(S_{(i_1,i_2)} - S_{(i_1^p,i_2^p)}, N_r^0). \end{aligned} \tag{16}$$

Here i_1^p and i_2^p are the predicted class indices which satisfy $f(i_1^p, i_2^p) = r$. S_e is the class index of the prediction error, and modulo operation is used to make sure that $0 \le S_e < N_r^0$. Hence, we propose to code S_e instead of $S_{(i_1,i_2)}$. Since the assignment class $r = \Gamma_{1,0}(s,t)$ is not changed by this operation, the upper levels of the index hierarchy is not affected by this change. During decoding the original index pair is recovered as:

$$S_{(i_1,i_2)} = \text{mod}\,(S_e + S_{(i_1^p,i_2^p)}, N_r^0),$$

For the estimation of prediction indices, the function $g(r, \alpha_p)$ is chosen such that the index pair is in the same class, i.e. $f(i_1^p, i_2^p) = r$, and their angle is as close to α_p as possible. For circular assignment, these conditions are satisfied as follows:

$$\begin{aligned} i_1^p &= \left\lfloor r \sin(\alpha_p) + 0.5 \right\rfloor, \\ i_2^p &= \left\lfloor \sqrt{r^2 - i_1^p} + 0.5 \right\rfloor. \end{aligned} \tag{17}$$

For the max function, the following equations describe $g(r, \alpha_p)$;

$$(i_1^p, i_2^p) = \begin{cases} \left(\text{sign}(\cos(\alpha_p))r, \quad \left\lfloor r \tan(\alpha_p) + 0.5 \right\rfloor\right), & \text{if } |\tan(\alpha_p)| \le 1 \\ \left(\left\lfloor r \cot(\alpha_p) + 0.5 \right\rfloor, \quad \text{sign}(\sin(\alpha_p))r\right), & \text{else} \end{cases} \tag{18}$$

Figures 9(a) and (b) visualize these operations for circular and max assignment functions, respectively (only first quadrant is shown).

As for the predicted angle α_p, experiments show that histogram of the angular difference between the two neighboring class assignments is peaked at π. Therefore, we use $\alpha_p = \arctan(\Gamma_{0,0}(2s, t-1), \Gamma_{0,0}(2s+1, t-1)) + \pi$. Figure

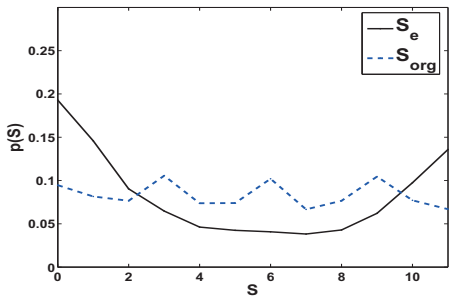


Figure 10: Histograms of $S_{(i_1,i_2)}$ and S_e, for $r = 2$ (*Lena* at 1 bpp).

10 shows the histograms of $S_{(i_1,i_2)}$ and S_e, for *Lena* image, when $r = 2$. We see that histogram of S_e is peaked around 0 and N_2^0, which indicates that the prediction is successful. The difference in entropy between two histograms in Figure 10 is 0.2, which is equivalent to 0.2 bits savings for each class assignment C_2^0.

It is worth mentioning that it would have been much more difficult to exploit any intra-band dependencies using the coding framework of the spherical tree. Since it is not possible to estimate the reconstructed values before the actual coding of the spherical tree, any predictions made during the construction of the tree could change during the encoding/decoding step, leading to bad reconstructions in rate-distortion sense. Moreover, rate-distortion optimization has to be carried out with the prediction errors and not with the actual wavelet coefficients, which leads to suboptimal tree constructions. The index tree representation avoids such problems by separating the coding of each tree level through the use of discrete-valued class assignments. Since the class indices are lossless coded, the predictions made at one level does not affect the coding of any other level.

6. Simulations

Hierarchical index coder is implemented using biorthogonal linear phase filter pairs in a 6-level dyadic decomposition. Same q and T are used in all subbands. Optimal q and T are chosen among the set $\{t : t = 0.1k,\ k = 50, 51, ..., 400\}$. Low-pass subband is arithmetic coded, after applying an (8×8) DCT, using optimal scalar quantizer for a given λ. Standard test images, *Lena*, *Goldhill* and *Barbara*, are used for simulations. The results are reported at 1.00, 0.50, 0.25 bits per pixel (bpp).

In Table 1, the performance of the wavelet-based hierarchical index coder, named as HIC, is compared to that of some of the best performing wavelet coders in the literature, including SPHE [14], SPIHT [3], SFQ [2], EBCOT [18], EZBC [6] and EQ [11]. In this first set of simulations, the index coder is using the circular assignment function.

As seen in Table 1, HIC outperforms SPHE, SPIHT, SFQ, and EZBC, when 9/7 filter pair is used. Except for *Barbara*, the performance of HIC is better than that of EBCOT, which is the algorithm used in JPEG2000 standard. Note that, EBCOT uses sophisticated contextual models which can adapt well to the local frequency content of textured regions in images such as *Barbara*. We believe that, with more advanced contextual modeling and adaptive coding

Table 1: PSNR comparison of different wavelet coders.

Lena		PSNR (dB)		
Coder	*Filter*	0.25 bpp	0.50 bpp	1.00 bpp
HIC	9/7	34.50	37.59	40.81
SPHE	9/7	34.28	37.40	40.67
SPIHT	9/7	34.11	37.21	40.46
SFQ	9/7	34.33	37.36	40.52
EBCOT	9/7	34.32	37.43	40.55
EZBC	9/7	34.35	37.47	40.62
HIC	10/18	34.57	37.65	40.83
EQ	10/18	34.57	37.69	40.88

Goldhill		PSNR (dB)		
Coder	*Filter*	0.25 bpp	0.50 bpp	1.00 bpp
HIC	9/7	30.89	33.55	37.02
SPHE	9/7	30.72	33.37	36.85
SPIHT	9/7	30.63	33.13	36.55
SFQ	9/7	30.71	33.37	36.70
EBCOT	9/7	30.75	33.38	36.75
EZBC	9/7	30.74	33.47	36.90
HIC	10/18	30.86	33.55	37.04
EQ	10/18	30.76	33.44	36.96

Barbara		PSNR (dB)		
Coder	*Filter*	0.25 bpp	0.50 bpp	1.00 bpp
HIC	9/7	28.52	32.35	37.31
SPHE	9/7	28.22	32.06	37.00
SPIHT	9/7	27.58	31.40	36.41
SFQ	9/7	28.29	32.15	37.03
EBCOT	9/7	28.53	32.50	37.38
EZBC	9/7	28.25	32.15	37.28
HIC	10/18	28.79	32.70	37.73
EQ	10/18	28.71	32.70	37.65

Table 2: PSNR comparison of different wavelet packet coders.

Lena		PSNR (dB)		
Coder	*Filter*	0.25 bpp	0.50 bpp	1.00 bpp
WP&HIC	9/7	34.55	37.56	40.75
WP&SPHE	9/7	34.40	37.44	40.65
WP&SFQ	9/7	34.35	37.40	40.55
WP&HIC	10/18	34.65	37.65	40.83
WP-EQ	10/18	34.49	37.66	40.80

Goldhill		PSNR (dB)		
Coder	*Filter*	0.25 bpp	0.50 bpp	1.00 bpp
WP&HIC	9/7	31.02	33.65	37.06
WP&SPHE	9/7	30.92	33.57	36.98
WP&HIC	10/18	30.98	33.59	37.04
WP-EQ	10/18	30.95	33.61	37.11

Barbara		PSNR (dB)		
Coder	*Filter*	0.25 bpp	0.50 bpp	1.00 bpp
WP&HIC	9/7	29.72	33.45	38.00
WP&SPHE	9/7	29.60	33.35	37.91
WP&SPIHT	9/7	29.00	32.73	37.34
WP&SFQ	9/7	29.37	33.13	37.70
WP&HIC	10/18	30.05	33.78	38.31
WP-EQ	10/18	30.00	33.87	38.51

tools, HIC could easily outperform EBCOT for *Barbara* as well. For 10/18 filter, HIC has better PSNR performance than EQ for *Goldhill* and *Barbara*; EQ is slightly better for *Lena*.

Simulations show that the performance of HIC improves by 0.1-0.3 dB by the use of context-adaptive coding techniques introduced in Section 5 for exploiting intra- and inter-band dependencies. For *Barbara*, the coding gains are around 0.2-0.3 dB, which confirms our belief in the use of better contextual models for further improving the coding efficiency.

HIC is also tested in wavelet packet coding of the same set of images. Best wavelet packet basis is coded as side information in the form of a pruned quadtree. The coding cost of this quadtree is less than 200 bits for all tested images. In Table 2, the performance of HIC in wavelet packets (i.e. WP&HIC) is compared to that of the wavelet packet versions of some of the previously mentioned coders, such as WP&SPHE [19], WP&SPIHT [7], WP&SFQ [7], WP-EQ [14]. All results are for the 9/7 filter pair, except for WP-EQ which uses the 10/18 filter. The index coder is using the circular assignment function. Once again, WP&HIC is better than WP&SPHE, WP&SPIHT and

Table 3: PSNR results for HIC and WP&HIC using circular and max assignment functions.

HIC	PSNR (dB)					
	Lena		*Goldhill*		*Barbara*	
Rate (bpp)	Max	Circ.	Max	Circ.	Max	Circ.
1.00	40.79	40.81	36.99	37.02	37.33	37.31
34.50 0.50	37.58	37.59	33.54	33.55	32.39	32.35
0.25	34.50	34.50	30.89	30.89	28.51	28.52

WP&HIC	PSNR (dB)					
	Lena		*Goldhill*		*Barbara*	
Rate (bpp)	Max	Circ.	Max	Circ.	Max	Circ.
1.00	40.73	40.75	37.04	37.06	37.98	38.00
0.50	37.55	37.56	33.63	33.65	33.44	33.45
0.25	34.54	34.55	31.02	31.02	29.73	29.72

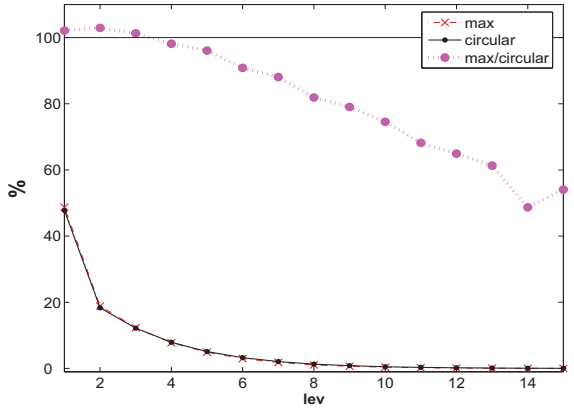


Figure 11: Comparison of bitrates for max and circular assignments (highest frequency bands of *Lena* at 1 bpp).

WP&SFQ, and almost as good as WP-EQ when same filter pair is used. For *Barbara*, 0.6-1.2 dB improvement over HIC shows the importance of using the best wavelet packet representation especially for textured images.

Compared to HIC, WP&HIC does not gain much from the use of intra- and inter-band neighbors for context adaptive coding. There are two reasons for this. First of all, due to the image-dependent decomposition of high-pass subbands in wavelet packets, there is no well-defined parent-children relationship among subbands in different scales. Also, wavelet packet decomposition tends to decorrelate high-pass subbands of the wavelet transform, resulting in much less dependency among intra-band neighboring coefficients. As a result, for *Lena* at 0.5 and 1.0 bpp, HIC outperforms WP&HIC, meaning that wavelet transform should actually have been selected as the best basis.

Next, the performance of max assignment function is evaluated. Table 3 shows PSNR results of HIC and WP&HIC using both max and circular assignment functions. With max function, PSNR is almost the same as that of circular

assignment for all the tested sequences. Figure 11 plots the percentage of bitrate spent by HIC at different levels of the index hierarchy (lev = $u+v$) for the highest frequency subbands of *Lena* at 1 bpp. The dotted line shows the ratio of the bitrates for max assignment and circular assignment at different levels. Since $\max(i_1, i_2) \leq \left\lfloor \sqrt{i_1^2 + i_2^2} + 0.5 \right\rfloor$, the class assignment values $\Gamma_{u,v}(s,t)$ of max function become increasingly smaller at higher levels of the hierarchy. Therefore, the bitrate spent at upper levels are comparably smaller (i.e. the ratio is less than 1) when max class assignment is used. On the other hand, since N_r for max assignment is generally greater than N_r for circular assignment at equal values of r (see Figures 4(a) and (b)), a higher bitrate is spent to code the lowest levels of the max assignment hierarchy, where the assignment values are more or less similar for both functions. In other words, max function moves the uncertainty in quantization indices from higher levels to the lower levels of the hierarchy, and the overall bitrate stays about the same.

As for the computational cost of the algorithm, the most time consuming part is to find the optimal parameter set (q, T, λ) for a target bitrate. Due to this exhaustive optimization, the complexity of the algorithm is comparable to that of SPHE, SFQ and EQ, and quite higher than the other algorithms mentioned above. However, our experiments indicate that the relationships between the parameters q, T, and λ can be reliably modeled, and a look-up table can be used for (q, T, λ), with less than 0.05 dB loss in PSNR for all tested cases.

Once the parameters (q, T, λ) are determined, the computational complexity of the coding procedure is reasonable. For building the hierarchy, the cost calculations require simple addition and comparison operations at each node. Encoding/decoding is performed using table look-up for class assignments. A significant portion of the coding complexity is due to context-based arithmetic coding of class assignments.

7. Conclusion

In this paper, we have introduced the quantization index hierarchy as a convenient and flexible data structure for classifying and coding wavelet and wavelet packet coefficients based on their quantization levels. This index tree is optimized for rate-distortion efficiency and coded hierarchically using two different class assignment functions. Intra- and inter-band dependencies of wavelet coefficients are also exploited through the use of context-adaptive coding techniques. The competitive results attained by the index coder point towards the potential of such hierarchical descriptions in coding wavelet subbands.

In future, we plan to investigate other class assignment functions, and try more sophisticated contextual models. In particular, class assignment function could also be adaptively selected, based on the contextual information. Also, wavelet coefficients could be further partitioned into multiple subsets depending on the type of assignment function chosen for optimal coding efficiency. We believe that wavelet coefficients have complicated intra- and inter-band dependencies that require more advanced coding tools, and the coding hierarchy of the index tree provides the right framework for developing such tools.

Hierarchical index coder, as described in this paper, is a non-progressive lossy coding method. Our current work

also focuses on different versions of hierarchical coder for lossless coding and for progressive coding by modifying the way in which the index tree is defined and coded.

References

[1] J. Shapiro, Embedded image coding using zerotrees of wavelet coefficients, IEEE Trans. Signal Processing 41 (12) (1993) 3445–3462.

[2] Z. Xiong, K. Ramchandran, M. Orchard, Space-frequency quantization for wavelet image coding, IEEE Trans. Image Processing 6 (5) (1997) 677–693.

[3] A. Said, W. Pearlman, A new fast and efficient image codec based on set partitioning in hierarchical trees, IEEE Trans. Circuit Syst. Video Tech. 6 (1996) 243–250.

[4] W. C. S. H. Pan, N. F. Law, Efficient and low-complexity image coding with the lifting scheme and modified SPIHT, in: Proc. IEEE Int. Conf. Neural Networks, Hong Kong, 2008, pp. 1959–1963.

[5] S. Chang, L. Carin, A modified SPIHT algorithm for image coding with a joint mse and classification distortion measure, IEEE Trans. Image Processing 15 (3) (2006) 713–725.

[6] S. T. Hsiang, Embedded image coding using zeroblocks of subband/wavelet coefficients and context modeling, in: Proc. Data Compression Conference (DCC '01), Washington, 2001, pp. 83–92.

[7] Z. Xiong, K. Ramchandran, M. Orchard, Wavelet packet image coding using space-frequency quantization, IEEE Trans. Image Processing 7 (6) (1998) 892–898.

[8] N. Sprljan, S. Grgic, M.Grgic, Modified SPIHT algorithm for wavelet packet image coding, Real-Time Imaging 11 (5-6) (2005) 378–388.

[9] Y. Yang, C. Xu, A wavelet packet based block-partitioning image coding algorithm with rate-distortion optimization, in: Proc. IEEE Int. Conf. Image Processing, Vol. 3, Genoa, Italy, 2005, pp. 201–204.

[10] R. Joshi, H. Jafarkhani, *et al.*, Comparison of different methods of classification in subband coding of images, IEEE Trans. Image Processing 6 (1997) 1473–1486.

[11] S. M. LoPresto, K. Ramchandran, M. T. Orchard, Image coding based on mixture modeling of wavelet coefficients and a fast estimation-quantization framework, in: Proc. Data Compression Conf., Snowbird, UT, 1997, pp. 221–230.

[12] M. K. Mihcak, K. Ramchandran, P. Moulin, Adaptive wavelet packet image coding using an estimation-quantization framework, in: Proc. IEEE Int. Conf. Image Processing, Chicago, 1998.

[13] H. F. Ates, M. T. Orchard, Wavelet image coding using the spherical representation, in: Proc. IEEE Int. Conf. Image Processing, Vol. 1, Genova, 2005, pp. 89–92.

[14] H. F. Ates, M. T. Orchard, Spherical coding algorithm for wavelet image compression, IEEE Trans. Image Processing 18 (5) (2009) 1015–1024.

[15] H. F. Ates, E. Tamer, Wavelet-based image compression by hierarchical quantization indexing, in: Proc. European Signal Process. Conf., Glasgow, 2009.

[16] R. Coifman, Y. Meyer, S. Quake, M. Wickerhauser, Signal processing and compression with wave packets, in: Proc. International Conference on Wavelets, Marseille, 1989.

[17] Y. Shoham, A. Gersho, Efficient bit allocation for an arbitrary set of quantizers, IEEE Trans. Acoust., Speech, Signal Processing 36 (9) (1988) 1445–1453.

[18] D. Taubman, High performance scalable image compression with EBCOT, IEEE Trans. Image Processing 9 (7) (2000) 1219–1235.

[19] H. F. Ates, E. Tamer, Spherical representation in wavelet packets for image coding, in: Proc. IEEE Conf. Signal Processing Communications Applications, Didim, 2008.