WAVELET-BASED IMAGE COMPRESSION BY HIERARCHICAL QUANTIZATION INDEXING

Hasan F. Ates, Engin Tamer

Dept. of Electronics Engineering
Isik University
Sile, Istanbul, TURKEY
hfates@isikun.edu.tr, engin.tamer@isik.edu.tr

ABSTRACT

In this paper, we introduce the quantization index hierarchy, which is used for efficient coding of quantized wavelet 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. Despite its simplicity, the algorithm produces PSNR results that are competitive with the state-of-art coders in literature.

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.

All of these successful wavelet 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 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 coefficients are labeled as significant and coded using scalar quantization. In other coders [7], 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 [8] makes use of local variance estimate to model the local statistics for more accurate bit allocation and superior coding efficiency. In spherical coding algorithm [9, 10], 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 introduce a wavelet-based coding algorithm that is based on a 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 [9], 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.

Section 2 introduces the hierarchical classification concept and describes the hierarchical index tree. Then, Section 3 explains the details of the coding algorithm based on the quantization index hierarchy. In Section 4, the performance of the coding algorithm is evaluated by using two different classification strategies and in comparison to the coding efficiency of some of the state-of-art wavelet coders.

This research was supported by Isik University BAP-05B302 Grant.

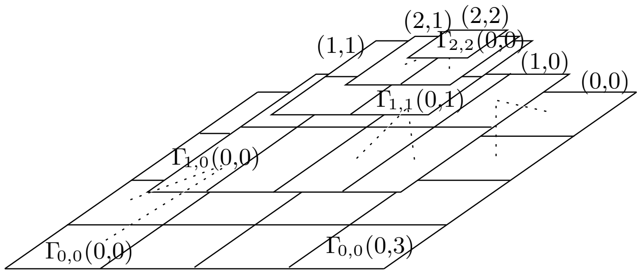

Figure 1: Hierarchical index tree ($J = 2$).

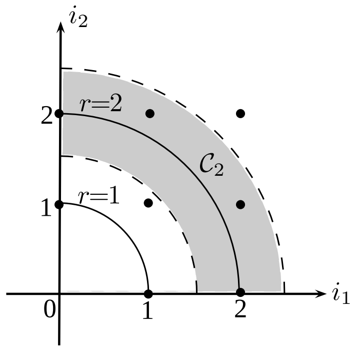

Figure 2: Circular assignment classes.

2. HIERARCHICAL QUANTIZATION INDEX CLASSES

Suppose that $c(m,n)$ represents the coefficients of the wavelet subband W_k^l, $1 \le k \le K$, of K level wavelet transform of a given image. Here, $l \in \{HL, LH, HH\}$ stands for the horizontal, vertical and diagonal subbands. Suppose W_k^l is of size $2^J \times 2^J$; then, $0 \le m,n < 2^J$. The absolute value of each coefficient, i.e. $|c(m,n)|$, is scalar quantized and assigned to a non-negative quantization level/index $i(m,n)$, as follows:

$$\begin{aligned} i(m,n) &= Q\left[|c(m,n)|\right], && (1)\\ \tilde{c}(m,n) &= \text{sign}(c(m,n))Q^{-1}\left[i(m,n)\right]. && (2) \end{aligned}$$

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

Now, we pair different quantization indices to form the following index classes:

$$\mathcal{C}_r = \{(i_1,i_2) | f(i_1,i_2) = r, \forall i_1, i_2, r \in \mathcal{Z}^+\} \quad (3)$$

where i_1, i_2, r are non-negative integers and $f(.,.)$ represents a class assignment function. The number of integer pairs in each class is defined to be $N_r = |\mathcal{C}_r|$. The assignment function is supposed to classify similar index pairs under the same class. Section 2.2 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. The first level of the hierarchical index tree is formed by pairing neighboring wavelet coefficients according to the class definition given above:

$$\begin{aligned} \Gamma_{0,0}(s,t) &= i(s,t),\\ \Gamma_{1,0}(s,t) &= f(\Gamma_{0,0}(2s,t), \Gamma_{0,0}(2s+1,t)). && (4) \end{aligned}$$

Likewise, the upper levels of the hierarchy are defined iteratively as follows (for $0 < u \le J$):

$$\begin{aligned} \Gamma_{u,u}(s,t) &= f(\Gamma_{u,u-1}(s,2t), \Gamma_{u,u-1}(s,2t+1)),\\ \Gamma_{u+1,u}(s,t) &= f(\Gamma_{u,u}(2s,t), \Gamma_{u,u}(2s+1,t)). && (5) \end{aligned}$$

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 1 shows some of the class assignment variables at different levels of the index hierarchy. The details of the coding procedure will be explained in Section 3.

2.1 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;

$$Q[c] = \begin{cases} 0 & \text{if } 0 \le c < T \\ \left\lfloor \frac{c-T}{q} + 1 \right\rfloor & \text{if } T \le c \end{cases} \quad (6)$$

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

$$Q^{-1}[i] = \begin{cases} 0 & \text{if } i = 0 \\ iq + T - q/2 & \text{else} \end{cases} \quad (7)$$

2.2 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 function 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 2):

$$f(i_1,i_2) = \left\lfloor \sqrt{i_1^2 + i_2^2} + 0.5 \right\rfloor \quad (8)$$

Hence, at each level of the hierarchy, $\Gamma_{u,v}(s,t)/2^{u+v}$, as defined in Eqn. (5), will approximately represent the root mean square of the quantization indices of underlying coefficients. In Figure 2, the gray area shows the class $\mathcal{C}_2$ and the dots in this area shows 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:

$$f(i_1,i_2) = \max(i_1,i_2) \quad (9)$$

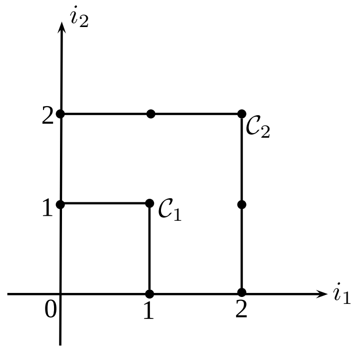

Figure 3: Max assignment classes.

This time $\Gamma_{u,v}(s,t)$ will correspond to the maximum index of the underlying coefficients. Figure 3 shows the classes $\mathcal{C}_r$, $r = 1, 2$ and the corresponding index pairs ($N_k = 2k + 1$) for this "max" assignment function. We will compare the performance of these two assignment functions in Section 4.

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 3 describes a simple optimization and coding strategy for efficient coding of the quantization index hierarchy.

3. WAVELET CODING BY QUANTIZATION INDEX HIERARCHY

The coding algorithm described in this section is applied independently in each wavelet subband, W_k^l, 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. 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. $\Gamma_{0,0}(m,n)$.

During encoding, at level (u,u) of the index hierarchy, given $\Gamma_{u,u}(s,t)$, we know from Eqn.(5) 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 $\mathcal{C}_{\Gamma_{u,u}(s,t)}$. 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 $\mathcal{C}_0$ (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 $\mathcal{C}_0$ is essential for improving the coding efficiency of the proposed algorithm.

Indeed, it turns out that building the hierarchical index tree using the original index values 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.

For that purpose, we propose a rate-distortion cost analysis that is similar to the one described in [9]. 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 (See Figure 1). 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, compare the Lagrangian cost of sending class variables of the wavelet coefficients to the cost of quantizing them all to zero. If the latter cost is smaller, then assign that subtree to $\mathcal{C}_0$. Suppose $L_{u,v}(s,t)$ represents the Lagrangian cost. 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.

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

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 Eqn.(5), and define the Lagrangian costs:

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

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

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

- 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 step 2.

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 $\Gamma_{u,u-1}(s,2t+1)$.

Table 1: PSNR comparison of different coders.

Lena	PSNR (dB)					
Rate (bpp)	HIC	SPHE	SPIHT	SFQ	EBCOT	EZBC
1.00	40.70	40.67	40.46	40.52	40.55	40.62
0.50	37.45	37.40	37.21	37.36	37.43	37.47
0.25	34.37	34.28	34.11	34.33	34.32	34.35

Goldhill	PSNR (dB)					
Rate (bpp)	HIC	SPHE	SPIHT	SFQ	EBCOT	EZBC
1.00	36.90	36.85	36.55	36.70	36.75	36.90
0.50	33.45	33.37	33.13	33.37	33.38	33.47
0.25	30.80	30.72	30.63	30.71	30.75	30.74

Barbara	PSNR (dB)					
Rate (bpp)	HIC	SPHE	SPIHT	SFQ	EBCOT	EZBC
1.00	37.05	37.00	36.41	37.03	37.38	37.28
0.50	32.14	32.06	31.40	32.15	32.50	32.15
0.25	28.34	28.22	27.58	28.29	28.53	28.25

- For $0 \leq s < 2^{(J-u)}, 0 \leq t < 2^{(J-u+1)}$, 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 step 3.

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 [11].

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. The coding model of each class $\mathcal{C}_{\Gamma_{u,v}(s,t)}$ is adapted based on the corresponding number of index pairs, $N_{\Gamma_{u,v}(s,t)}$, and the level of the tree, i.e. (u,v) pair. Note that, the output bitrate of the arithmetic coder turns out to be only slightly better than the entropy estimate assuming equally probable index pairs. In other words, it is justified to use the self-information $\log_2(N_{\Gamma_{u,v}(s,t)})$ for estimating the bitrate required to code class assignments.

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.

4. SIMULATIONS

Hierarchical index coder is implemented using 9/7 biorthogonal linear phase filter pair in a 6-level dyadic decomposition. Same q and T are used in all subbands. Optimal q and T are chosen among the set $\{t : t = 0.1k\pi, k = 1, 2, ..., 150\}$. Low-pass subband is arithmetic coded, after applying an (8×8) DCT, using optimal scalar quantizer for a given λ.

In Table 1, the performance of the index coder is compared to that of some of the best performing coders in the literature, including SPHE [10], SPIHT [3], SFQ [2], EBCOT [12] and EZBC [6]. *Lena*, *Goldhill* and *Barbara* images are used for comparison. All results are for the 9/7 filter pair. In this first set of simulations, the index coder is using the circular assignment function.

The hierarchical index coder, called as HIC in Table 1, outperforms SPHE, SPIHT and SFQ, and is as good as EBCOT and EZBC in most cases. 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*. Considering the simplicity of the coding choices we have made in the index coder, these results are rather encouraging for our future efforts in developing highly efficient and adaptive coding methods based on the index assignment functions.

Table 2: PSNR results for HIC using circular and max assignment functions.

HIC	PSNR (dB)					
	Lena		*Goldhill*		*Barbara*	
Rate (bpp)	Max	Circ.	Max	Circ.	Max	Circ.
1.00	40.69	40.70	36.88	36.90	37.08	37.05
0.50	37.45	37.45	33.44	33.45	32.17	32.14
0.25	34.37	34.37	30.80	30.80	28.35	28.34

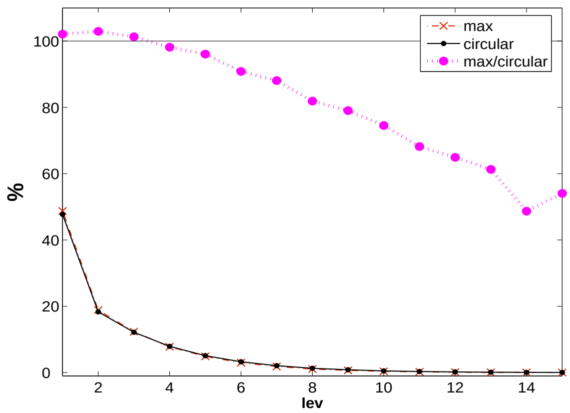

Figure 4: Comparison of bitrates for max and circular assignments (highest frequency bands of *Lena* at 1 bpp).

Next, the performance of max assignment function is evaluated. Table 2 shows PSNR results of 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 4 plots the percentage of bitrate spent 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 2 and 3), 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, the 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.

5. CONCLUSION

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

The computational complexity of the algorithm is low, for a given set of parameters (q, T, λ). 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. We are currently working on modeling the relationships between the parameters q, T, and λ, to estimate their values without needing any exhaustive search.

In future, we plan to investigate other class assignment functions, and try adaptive selection of class assignments at different hierarchy levels and by using inter-scale dependencies between neighboring wavelet subbands. In particular, wavelet coefficients could be further partitioned into multiple subsets depending on the type of assignment function chosen for optimal coding efficiency.

REFERENCES

[1] J. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3445–62, Dec. 1993.

[2] Z. Xiong, K. Ramchandran, and M. Orchard, "Space-frequency quantization for wavelet image coding," *IEEE Trans. Image Processing*, vol. 6, no. 5, pp. 677–93, May 1997.

[3] A. Said and W. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuit Syst. Video Tech.*, vol. 6, pp. 243–50, June 1996.

[4] W. C. S. H. Pan and 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 and L. Carin, "A modified SPIHT algorithm for image coding with a joint mse and classification distortion measure," *IEEE Trans. Image Processing*, vol. 15, no. 3, pp. 713–725, Mar. 2006.

[6] S. 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] R. Joshi, H. Jafarkhani, and *et al.*, "Comparison of different methods of classification in subband coding of images," *IEEE Trans. Image Processing*, vol. 6, pp. 1473–86, Nov. 1997.

[8] S. M. LoPresto, K. Ramchandran, and M. T. Orchard, "Image coding based on mixture modeling of wavelet coefficients and a fast estimation-quantization framework," in *Proc. Data Compression Conf.*, Snowbird, UT, March 1997, pp. 221–30.

[9] H. Ates and M. Orchard, "Wavelet image coding using the spherical representation," in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, Genova, Sept. 2005, pp. 89–92.

[10] ——, "Spherical coding algorithm for wavelet image compression," *accepted to IEEE Trans. Image Processing*, 2009.

[11] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, no. 9, pp. 1445–53, Sept. 1988.

[12] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Trans. Image Processing*, vol. 9, no. 7, pp. 1219–35, July 2000.