

Vector Quantization for Efficient Coding of Upper Subbands<sup>1</sup>

W.J. Zeng  
 Department of Electrical Engineering  
 Princeton University  
 Princeton, N.J. 08544  
 e-mail: wzeng@ee.princeton.edu

Y.F. Huang  
 Lab. Image & Signal Analysis  
 Department of Electrical Engineering  
 University of Notre Dame  
 Notre Dame, IN 46556  
 e-mail: huang.2@nd.edu

## Abstract

This paper examines the application of vector quantization (VQ) to exploit both intra-band and inter-band redundancy in subband coding. The focus here is on the exploitation of inter-band dependency. It is shown that VQ is particularly suitable and effective for coding the upper subbands. Three subband decomposition-based VQ coding schemes are proposed here to exploit the inter-band dependency by making full use of the extra flexibility of VQ approach over scalar quantization. A quadtree-based variable rate VQ (VRVQ) scheme which takes full advantage of the intra-band and inter-band redundancy is first proposed. Then, a more easily implementable alternative based on an efficient block-based edge estimation technique is employed to overcome the implementational barriers of the first scheme. Finally, a predictive VQ scheme formulated in the context of finite state VQ is proposed to further exploit the dependency among different subbands. The VRVQ scheme proposed in [29] is extended to provide an efficient bit allocation procedure. Simulation results show that these three hybrid techniques have advantages, in terms of peak signal-to-noise ratio (PSNR) and complexity, over other existing subband-VQ approaches [1, 2, 18, 21, 27].

(NASA-CR-196804) VECTOR  
 QUANTIZATION FOR EFFICIENT CODING  
 OF UPPER SUBBANDS (Notre Dame  
 Univ.) 27 p

N95-11762

Unclass

G3/61 0022364

<sup>1</sup>This work was done while W.J. Zeng was with Laboratory for Image and Signal Analysis, Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556. This work has been supported, in part, by NASA Lewis Research Center under Grant NAG3-1549.

NIS  
 11-6-80  
 p. 4/27

# 1 Introduction

Among a variety of image coding methods, vector quantization (VQ) and subband coding (SBC) have attracted considerable interest. This is mainly due to the facts that VQ can, in theory, always achieve better performance than scalar quantization (SQ) [3, 6, 22], and SBC has been shown to achieve high compression ratios while maintaining good image visual quality [1, 8, 28]. Much work has been done on combining these two methods. While many schemes use VQ to exploit the intra-band redundancy [1, 18, 21], little is done to exploit the inter-band redundancy with some exceptions like those of [15, 27].

The advantages of VQ over SQ have been discussed elaborately in [6]. One of the disadvantages of VQ is that it often results in blocky images like other block-based coding schemes do. Over the last many years, various VQ techniques which aim to improve the visual quality of the reconstructed image have been developed. Typical such techniques include classified VQ [17], variable rate VQ (VRVQ) [20, 29], subband-VQ [1, 27], etc.. This paper examines the application of VQ in the frequency domain and shows that the subband-VQ approach is indeed viable for image compression.

The employment of subband decomposition for image coding was first reported in the work of Woods and O'Neil [28]. Since then much work has been devoted to subband image coding, see, e.g., [1, 2, 7, 8]. The essential idea of subband coding is that of analyzing the original fullband signal into a set of narrowband signals that can be encoded separately and then transmitted over communication channels. At the receiver end, the decoded subbands will then be used to synthesize an approximation to the original signal.

It is often desirable to design analysis and synthesis filters to meet the perfect reconstruction property [25]. One of the most popular perfect reconstruction filters is the quadrature mirror filter (QMF) introduced by Croisier [4]. Figure 1 shows a typical pyramidal subband decomposition for image coding [8]. At the first stage of decomposition, the original image is decomposed by using QMF's into four bands denoted by  $LL$ ,  $LH$ ,  $HL$ , and  $HH$ . Then all four bands are decimated by a factor of 4 and the lowest band  $LL$  is again decomposed into four bands by QMF's. Recursive applications of this procedure lead to the pyramidal decomposition of desired stages. By reversing the decomposition procedure and using the synthesis filter banks, one can reconstruct the image perfectly.

Recently, subband approaches using wavelet transforms [5] have been investigated by many researchers, see, e.g., [1]. As discussed in [14], the QMF's are intimately connected with wavelet transform. In this paper, the *Daubechies* wavelet analysis/synthesis filter banks [5] are used to perform the subband decomposition.

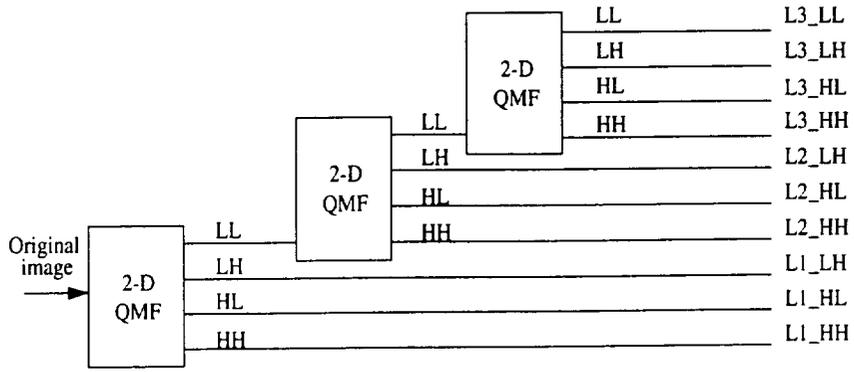


Figure 1: A pyramidical subband decomposition

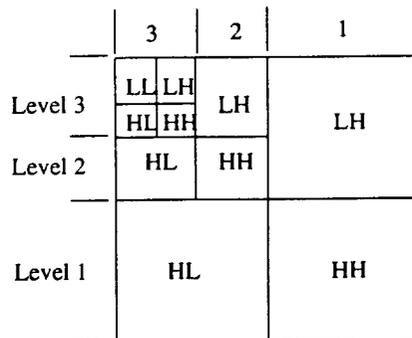


Figure 2: A three-stage wavelet decomposition

The coding system is based on a three-stage decomposition as shown in Fig. 2. Ten different image subbands are generated and are denoted by  $L1\_LH$ ,  $L1\_HL$ ,  $L1\_HH$ ,  $L2\_LH$ ,  $L2\_HL$ ,  $L2\_HH$ ,  $L3\_LH$ ,  $L3\_HL$ ,  $L3\_HH$ ,  $L3\_LL$ , as shown in Fig. 1.

The statistical properties and perceptual features of subband images have been discussed in [13, 24, 30]. These properties, if exploited effectively, can result in significant performance improvement in image coding. Among these properties, the self-similarity among subbands of "like" orientation has attracted much attention recently [11, 15, 27]. Though uncorrelated or "almost" uncorrelated, subbands at "like" orientations and different scales (resolutions) are generally not independent, since these subband signals are generally not Gaussian distributed. Figure 3 shows the subband images of "Lenna" after a three-stage wavelet decomposition. It is clear from this picture that subbands at "like" orientations and different scales look very similar. Refer to Level 1 and Level 2 of this decomposition as the upper bands. One may view the upper bands of this decomposition as having the characteristics of the output of a multiscale edge detector [2]. The coefficients at Level 1 may be considered points of very fine edges, while the coefficients at Level 2 may be considered points of coarser edges.

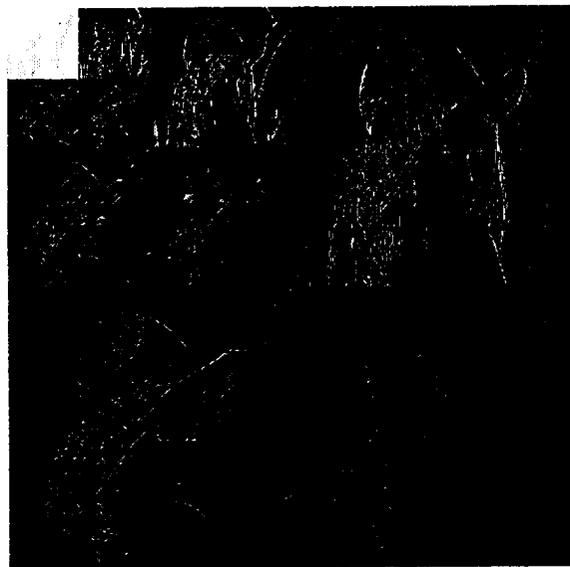


Figure 3: Similarity within sets of subbands

A number of image coding techniques [11, 15, 27] have exploited the redundancy among subbands, namely, the inter-band dependency. Johnsen et al. [11] used a local measure of activity in the base-band to classify the intensity of the samples in the upper frequency bands. The frequency components of the

upper bands are then divided into different groups where optimal scalar quantizer is designed for each group. Recently, Mohsenian and Nasrabadi [15] extended this idea to image and video coding using an edge-based VQ technique for compression of the upper bands. Similarly, Banham and Sullivan [2] suggested a scheme that utilized the multiscale edge characteristics of the wavelet transform to describe the relationship among coefficients at different scales. In [2], each coefficient in the subbands was considered as corresponding to a block of pixels in the original image. For example, in our 3-stage wavelet decomposition system (Fig. 2), a coefficient in Level 3 corresponds to an  $8 \times 8$  block in the original image. These coefficients can be structured into quadtrees for each  $8 \times 8$  block in the original image [2]. Figure 4 (redrawn from [2]) shows the relationship between blocks in the original image and coefficients in a three-stage wavelet decomposition. A sample quadtree is also shown, demonstrating how the coefficients of different levels fall naturally into a quadtree structure. In [2], edges in Level 1 were used to predict the location and intensity of edges in Level 2. The coarser edge is considered as a blurred version of the corresponding finer edges over a region which is twice as large in each dimension. This fact is used to predict a coefficient of the wavelet transform at Level 2 as a linear combination of four corresponding coefficients at Level 1 plus some error term, thereby reducing the average bit rate as in DPCM coding.

In general, VQ is a good technique to exploit inter-pixel correlations in the lowest subband. Even though pixels in higher subbands are usually not highly correlated, VQ can still offer better performance over SQ due to its flexibility in choosing the cell shape. More importantly, the existence of nonlinear dependency between different subbands challenges the conventional scalar quantization schemes. This nonlinear dependency is difficult to be thoroughly investigated with such scalar quantization schemes as DPCM and PCM. Generalizing SQ, VQ can offer more flexibility and thus may be an efficient way to solve this problem.

This paper studies the application of VQ in the wavelet transform domain. Specifically, three VQ-based subband coding schemes are proposed to exploit both the intra-band and inter-band redundancy, especially the self-similarity property of subband images. The main objective of these three schemes is to exploit the inter-band redundancy to the furthest extent. The first scheme takes advantage of the edge feature separation property of subband decomposition and employs a product-code-like VQ approach to fully exploit the inter-band redundancy. The second scheme aims to reduce the complexity of the first one by using a small vector dimension, while reducing the bit rate by eliminating non-edge information. Taking advantage of the shape and gain feature of VQ approach, the third one aims to further reduce the inter-band redundancy. It will be shown that VQ is indeed quite suitable and effective for coding upper subbands of a wavelet decomposition.

It will also be seen that the jump from one dimension to multiple dimension stimulates some new ideas, concepts and techniques that often have no counterpart in the simple case of scalar quantization [6].

This paper is organized as follows: Section 2 introduces an efficient block-based edge estimation technique to eliminate the "non-edge" information. Section 3 presents a quadtree-based variable rate VQ (VRVQ) scheme which takes full advantage of the intra-band and inter-band redundancy. In Section 4, a more easily implementable alternative based on the efficient block-based edge estimation technique is employed to overcome the implementational barriers of the scheme proposed in Section 3. Finally, a finite state VQ technique is proposed in Section 5 to further exploit the dependency among different subbands. Experimental results are also shown in these sections to examine these proposed schemes in comparison to other existing ones. Section 6 concludes this paper.

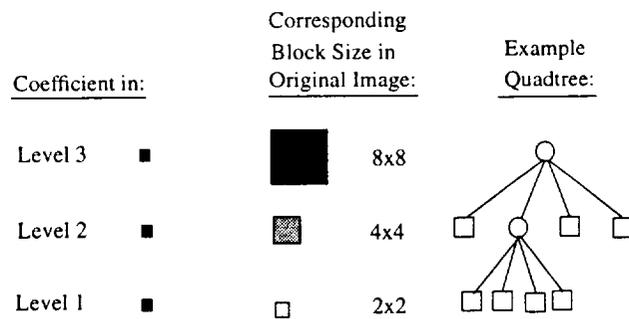


Figure 4: Coefficients vs. block size in quadtree

## 2 Prediction of Edge Locations from Coarser Scales

The methods used in [11, 15] to extract the locations of edges are based on detecting the edges in the base-band. These methods do not explicitly specify the orientations of the edges. Considering the fact that edges of "like" orientation appear in corresponding locations in "like" orientation subbands of different scales, it may be more efficient to estimate the edge locations in the current scale from the next coarser scale. In this way, the orientations of the edges can be specified.

Intuitively, one may use one pixel value of the next coarser scale to estimate the values of the corresponding four pixels in the current scale. For example, if the value of a pixel in the next coarser scale is less than a

threshold, one classifies the corresponding four pixels in the current scale as "non-edge" block and sets the values of all four pixels zero (Predict\_1.to\_4). This, however, is not a good scheme due to the downsampling nonlinearity. In particular, one pixel value of the next coarser scale may be very small while the corresponding  $2 \times 2$  block in the current scale contains edge information. An alternative is to interpolate the next coarser scale to the same size as the current scale, then use the interpolated value of one pixel to estimate the value of the corresponding pixel in the current scale (Predict\_1.to\_1). This is unfortunately not good either, as the edge locations in the interpolated sub-image are just approximations of the edge locations in the current scale. There may be one or two pixel units shifts.

We now propose a more accurate and reliable estimation scheme to deal with the problem. We still use the interpolated version of the next coarser scale to estimate the edge locations of the current scale. However, instead of using one pixel in the interpolated sub-image to estimate one pixel value in the current scale, our estimation is based on a  $2 \times 2$  block. If all the values in a  $2 \times 2$  block of the interpolated sub-image are below a threshold, we consider the corresponding  $2 \times 2$  block of the current scale as containing no edge information and set the values of all four pixels zero (Predict\_4.to\_4)<sup>2</sup>.

To examine this proposed scheme, simulations have been done on a brain image and the results of the three aforementioned methods are compared. We also show the results of performing the "deadzoning" by comparing the real value of one pixel of the current scale to a prescribed threshold (Same\_1.to\_1), and by comparing the real values of  $2 \times 2$  blocks of the current scale to a prescribed threshold (Same\_4.to\_4). Each of these five schemes is performed for all three scales. To make a fair comparison, for a fixed scale, the threshold for each scheme is chosen in such a way that the numbers of pixels classified as "non-edge" are almost the same for all schemes. Table 1 shows that the proposed scheme, i.e., Predict\_4.to\_4, yields better results than both Predict\_1.to\_4 and Predict\_1.to\_1 and is reasonably close to the result provided by Same\_1.to\_1. Note that in these cases, about 45% of the pixels are eliminated. This is done without any side information to be sent, since the edge location estimation is performed both at the transmitter end and the receiver end. The substantial saving of bit rate using a similar strategy has been discussed in [15]. In practice, the interpolated sub-image is based on the coded version of the next coarser scale. Therefore the proposed block-based method is more reliable than the pixel-based methods, because the quantization errors in the coded version of the next coarser scale have more severe effects on the pixel-based method than on the block-based method.

---

<sup>2</sup>It has been noticed recently that similar idea has been introduced in [12] where a  $2 \times 2$  block in the next coarser scale is used to predict the importance of the corresponding  $4 \times 4$  block in the current scale.

methods	MSE	No. of pels eliminated		
		Level 3	Level 2	Level 1
Predict_1_to_1	14.69	748	4560	23776
Predict_1_to_4	38.06	744	4572	23788
Predict_4_to_4	5.23	748	4568	23792
Same_4_to_4	2.46	744	4576	23788
Same_1_to_1	1.21	748	4576	23808

Table 1: A Comparison of different estimation methods for the "brain" image

### 3 Quadtree VRVQ for Upper Subbands

The VQ scheme for coding the upper subbands to be described here is based on the quadtree structure discussed in Section 1. The objective here is to exploit inter-band as well as intra-band redundancy. This may be accomplished by taking each quadtree in the decomposition as a vector. In the three-stage decomposition case, Fig. 4, one may take one pixel from Level 3, four pixels from Level 2, and sixteen pixels from Level 1. In fact, this quadtree structure naturally matches the perceptual features of the subbands. Note that the sub-blocks of a quadtree belonging to coarser scales, which need finer quantization, have a smaller block size which generally reduces the block effect in VQ.

It is interesting to note that this scheme is closely related to the product VQ technique [6]. Corresponding to each  $8 \times 8$  block in the original image, there are 64 coefficients in the transformed domain. Specifically, there are three quadtrees each of which consists of 21 coefficients, for the vertical, horizontal and diagonal direction respectively, plus one coefficient from the lowest frequency band. If one takes these 64 coefficients to form a vector and uses a 64-dimensional vector quantizer to code it, the performance in terms of PSNR is unlikely to be superior to that of directly coding the  $8 \times 8$  block in the original image using a 64-dimensional vector quantizer. However, the point here is that one need not take the 64 coefficients to form a vector of "large" dimension. Observe that the information of each  $8 \times 8$  block in the original image has been partitioned into 4 vectors of different features. If these four feature vectors are approximately independent of one another, then the coding complexity can be greatly reduced by coding them separately without a substantial performance degradation [6]. The coefficient from the lowest frequency band can be considered

as the normalized weighted sum of the samples of the original  $8 \times 8$  block and can often be regarded as statistically independent of the other three feature vectors. The other three feature vectors of dimension-21 represent shape characteristics for three different directions. They are not independent of one another in general. For example, if a horizontal edge feature vector (quadtree) contains horizontal edge, it is often true that the corresponding vertical edge feature vector will not contain an edge. Fortunately, as will be shown in the following, by applying the block-based edge estimation technique discussed in Section 2, these "non-edge" vectors will be eliminated without consuming extra bits. Therefore, coding these vectors of different features separately will not result in substantial performance degradation. Meanwhile, since the vector dimension is reduced and the features have been classified, the implementational complexity will be greatly reduced.

### 3.1. Estimation of Important Quadrees

Obviously, some of the quadrees contain edge information while others do not. Therefore the "non-edge" quadrees can be eliminated without incurring much distortion. It would be beneficial to use the edge information contained in the lowest band to extract the locations of important quadrees for each of the three orientations. This can be accomplished by using the block-based estimation method described in Section 2. Specifically, we use the analysis filter to filter the lowest band and obtain three filtered versions of sub-image of three different orientations. If all the values of the pixels in a  $2 \times 2$  block of the filtered versions of sub-image are below a threshold, the corresponding 4 quadrees are treated as "zero" quadrees.

In this case the block-based estimation method shows more advantages over pixel-based methods. Here the estimation accuracy is more important because one estimation error will affect the whole quadtree instead of only one pixel. Furthermore, if all the values of a  $2 \times 2$  block in one scale are very small, it is very likely that the values of the corresponding  $4 \times 4$  block in the next finer scale are very small, too. On the contrary, if one pixel in one scale is very small, it is often difficult to determine whether the values of the corresponding  $2 \times 2$  block in the next finer scale is small or not. Table 2 shows the reconstructed errors resulting from different estimation methods. The error images obtained by subtracting the reconstructed image from the original image are shown in Figure 5. The advantages of the block-based estimation method are obvious.

### 3.2. Bit Allocation

The quadrees can be naturally classified into three classes, with each class corresponding to one orientation. To preserve edges well, it is necessary to construct three sub-codebooks for these three classes. As

Methods	Predict_1_to_1	Predict_1_to_4	Same_1_to_1	Predict_4_to_4	Same_4_to_4
MSE	19.71	19.52	8.16	6.58	5.26
No. elim.	3602	3616	3602	3608	3604

Table 2: A comparison of different estimation methods for "Lenna" image

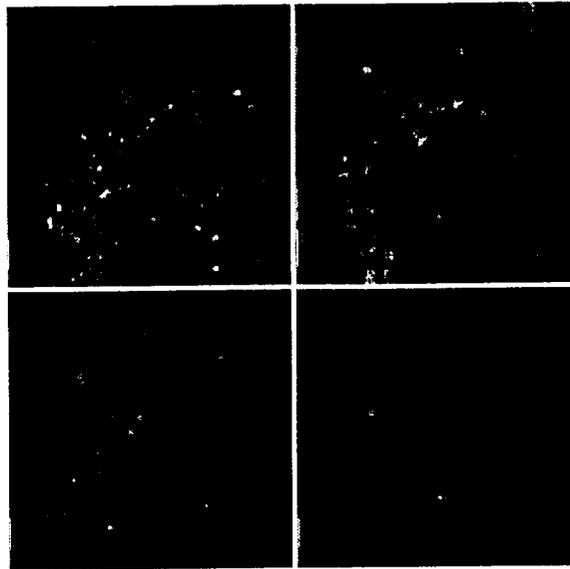


Figure 5: Error "Lenna" images resulting from different estimation methods. a: Predict\_1\_to\_1; b: Predict\_1\_to\_4; c: Same\_1\_to\_1; d: Predict\_4\_to\_4.

a	b
c	d

discussed in [24, 30], each orientation may have different characteristics and need different numbers of bits to code. Therefore an efficient bit allocation scheme should be employed here.

As discussed in [30], for a bit allocation problem with the subsource indices known by the receiver, the asymptotically optimal solution is the one that results in the same *average* distortion for different subsources. Using this as a guideline, we extend the variable rate VQ schemes proposed in [29] and propose the following procedure, referred to here as Greedy Bit Allocation Procedure (GBAP), for effectively allocating an average bit rate  $R$  bits per vector among subbands.

**Step 0** The entire space is partitioned into three subspaces, i.e.,  $v_1, v_2, v_3$ , one for each of the three orientations. Calculate the average distortions for the three subspaces.

**Step 1** Let  $v^*$  be the subspace that has the largest average distortion, and let  $\text{bitrate}=0$ ,  $L=3$ .

**Step 2** Calculate the mean  $y^*$  for  $v^*$ . Partition  $v^*$  into two regions by the hyperplane  $H = \{x : u^{*T}(x - y^*) = 0\}$ , where  $u^*$  is the principal component of  $v^*$  [9]. Let  $L=L+1$ ,  $\text{bitrate}=\text{bitrate}+p^*$ , where  $p^*$  is the probability of access of  $v^*$ .

**Step 3** If  $\text{bitrate} \geq R$ , stop. Otherwise, set the partition  $V_L = \{v_1, v_2, \dots, v_L\}$ . Select the region  $v^*$  with the largest average distortion for the next partition. Goto Step 2.

A typical tree for this procedure is depicted in Fig. 6. Note that the root node has three children, one for each orientation. The subtree rooted at each child is grown in the same way as in the variable rate VQ schemes proposed in [29]. The point here is that we allow the selection of the node to split to be performed interweavingly among the three orientations. Thus the bit allocation procedure can match the characteristics of the source as much as possible. This whole process will be referred to as subband Quadtree VRVQ Algorithm (SQVRVQA) in the sequel. Note that this scheme has little design and encoding complexity due to its simple structure [29].

### 3.3. Simulation Results

In this experiment our training set consists of ten  $512 \times 512$  black-and-white images, all of which are human pictures or natural pictures. The  $512 \times 512$  black-and-white "Lenna" image is used as the test image. Both inside test, i.e., the "Lenna" image is inside the training set, and outside test are performed. The 4-taps *Daubechies* wavelet filter [5] is used to obtain the wavelet decomposition.

After three stages of decomposition, the lowest subband of the "Lenna" image is coded by DPCM and

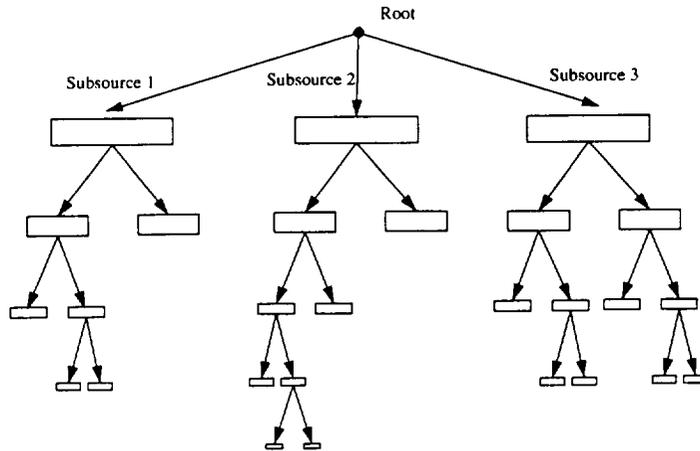


Figure 6: A typical tree structure for GBAP bit allocation procedure

Huffman coding [10]. To exploit both the vertical and horizontal pixel correlations, the neighboring three pixels  $A$ ,  $B$  and  $C$  as shown in Figure 7 are used for the prediction of the current pixel  $X$ . Then the quadtrees containing the edge information from the 10 training images form the training vectors. The codebook is generated using the GBAP procedure. Note that there are three sub-codebooks, one for each of the three orientations. The "edge quadtrees" of the "Lenna" image are coded using the codebook, and then reconstructed to form the full band coded image.

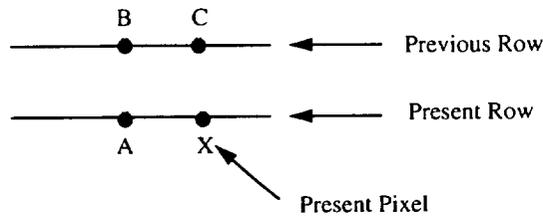


Figure 7: Configuration of pels used for prediction

The coding results are shown in Table 3 and Fig. 8. For the inside test, an excellent result with PSNR of 34.95 dB is obtained at the bit rate of as low as 0.366 bpp. For outside test, however, degradation can be observed and the PSNR also reduces to around 30 dB. But the visual quality is still good, compared to that of other non-subband VQ approaches (see, e.g., Figure 9). This seems to suggest that wavelet transform coding can provide better visual quality.

Method	Case	Bit Rate (bpp)			PSNR (dB)
		Lowest Band	Upper Bands	total	
SQVRVQA	Inside	0.076	0.290	0.366	34.95
	Outside	0.076	0.282	0.358	29.77
Ref. [27]	Inside			0.418	26.57
	Outside			0.418	25.73
Ref. [27]	Ideal			0.437	31.94

Table 3: Coding results for "Lenna" image by SQVRVQA and another subband-VQ scheme



Figure 8: SQVRVQA coded "Lenna" image for inside test (left, PSNR=34.95 dB, bpp=0.366) and outside test (right, PSNR=29.77 dB, bpp=0.358), filter: 4-taps.



Figure 9: Riskin-Gray VRVQ [20] coded "Lenna" image when using "Lenna" as the only training image. PSNR=30.10 dB, bpp=0.41.

In Table 3, the results obtained for the scheme proposed in [27] are based on using VQ to encode vectors of dimension 16, with one pixel from each of the 16 equally divided subbands. For the same training set of 10 images, the performance is much worse than that of SQVRVQA. Even when we use the test image as the only training image (ideal case in Table 3) for the scheme in [27], the result is still worse than that of SQVRVQA when the test image "Lenna" is inside the training set of 10 images. A possible reason for such performance differences between these two schemes is that SQVRVQA actually exploits the correlation of the  $8 \times 8$  blocks in the original image, while the scheme of [27] exploits the correlation of the  $4 \times 4$  blocks in the original image. It is well known that the larger the vector dimension is, the better the PSNR performance is in general. Of course, larger dimension often results in more complexity. However, in SQVRVQA, the vector dimension is reduced from 64 to 21. Therefore the complexity is not substantially increased.

The performance degradation for the outside test of SQVRVQA is probably due to the fact that the statistical characteristics of the test image do not match those of the training set well. This is an issue that often occurs in the VQ design problem. Generally speaking, to evaluate the performance of a VQ technique realistically, it is practically preferable to perform an outside test. However, if the characteristics of the test image do not match those of the training set well, this evaluation may not be fair. This problem is more pronounced when the vector dimension is large, since it is usually more difficult to find a set of

training vectors of a large dimension with consistent statistical characteristics. To improve the outside test performance of SQVRVQA, a larger and statistically more consistent training set is needed. This, however, calls for more memory and design complexity and thus may not be feasible.

## 4 An Alternative to Quadtree VRVQ for Upper Subbands

Last section showed a viable VQ technique for the coding of upper bands. However, its performance is limited by the memory requirement and codebook design complexity. This section proposes an efficient alternative with less complexity and storage requirement.

The practical limitation of the quadtree VRVQ is due to the large dimension of the quadtree vectors. If one codes subbands of different scales separately, the vector dimension can be reduced. This, however, loses the nice feature of the quadtree VQ scheme, i.e., the advantage of exploiting the inter-band dependency. But, as will be shown in the following, the performance loss can be minimized if one carefully employs the edge estimation technique (Predict\_4\_to\_4) for subbands of each scale to exploit the inter-band redundancy.

It has been shown in Section 2 that edge estimation based on a  $2 \times 2$  block is better than just one pixel. Therefore it is convenient and natural to use VQ to code the  $2 \times 2$  blocks which contain edge information. This alternative will be referred to as Block-based Edge Estimation VRVQ (BEEVRVQ) later. The BEEVRVQ has more flexibility than the quadtree VQ scheme, since one can use different deadzone thresholds for different scales. This implies that one can take full advantages of the perceptual properties of the subbands. Notice that even though a quadtree is classified as "edge quadtree", it is very likely that some sub-blocks of this quadtree, especially those belonging to the finer scale, contain no important information. In BEEVRVQ, one can choose appropriate thresholds to eliminate such sub-blocks.

### 4.1. Bit Allocation

The bit allocation procedure introduced in Section 3 can also be applied here. Here we have more than three subsources from different scales and different orientations. One can generate all the sub-codebooks concurrently as done in Section 3.

There is an "almost" equivalent alternative. Observe that the stopping criterion in the GBAP procedure of Section 3, "stop if the actual bit rate is greater than the prescribed bitrate (Criterion 1)", is almost equivalent to the criterion: "stop if the largest average distortion is less than a corresponding threshold

(Criterion 2)". It is "almost equivalent" because the prescribed bitrate and the final largest average distortion are not necessarily one-to-one corresponding to each other. This is because the largest average distortion is not necessarily monotonically decreasing during the growing process. Also, notice that "the largest average distortion for the entire tree is less than one threshold" implies that "the largest average distortion for each subtree of one subsource is "almost" just below the same threshold". Therefore one can set the threshold for the average distortion and use Criterion 2 in the construction of the tree, thus grows the subtree for each subsource separately. This may reduce the memory requirement and simplify the programming.

## 4.2. Simulation Results and Discussion

In this experiment our training set consists of the same ten  $512 \times 512$  black-and-white images as in Section 3. The  $512 \times 512$  black-and-white "Lenna" image is used as the test image.

After three stages of decomposition, the lowest frequency subband of the "Lenna" image is coded by DPCM and Huffman coding. To preserve better visual quality, the other three subbands in Level 3 are coded by PCM plus Huffman coding, rather than by VQ. For each subband in Level 1 and Level 2, the  $2 \times 2$  vectors containing the edge information from the 10 training images form the training sequences. The sub-codebooks for the three subbands in Level 1 are generated concurrently, while the sub-codebooks for the three subbands in Level 2 are generated concurrently. Then the "edge vectors" of the "Lenna" image are coded using the codebook, and reconstructed to form the full band coded image.

The coding results are shown in Table 4 and Fig. 10. It is interesting to observe that the PSNR's for both the inside test and outside test are very good and very comparable. Due to the reduced complexity and memory requirement, this scheme can provide high outside test PSNR which seems unlikely to be achievable in practice by the SQVRVQA scheme. Of course the price to pay here is a slightly higher bit rate.

It is clear from Table 4 and Fig. 10 that BEEVRVQ compares favorably to other existing schemes. In Table 4 we first show results for some other subband-VQ approaches. For  $512 \times 512$  "Lenna" image, Antonini et al. [1] reported good visual quality images of PSNR of 32.10 dB using VQ-based wavelet transform coding combined with entropy coding at 0.78 bpp. Westerink [27] obtained a PSNR of 32.0 dB at 0.63 bpp using subband-VQ. Safranek et al. [21] used multistage FSVQ to selectively quantize the reconstruction noise in the dominant subband and achieved a PSNR of 32.5 dB at 0.5 bpp. Mohsenian et al. [15] reported a PSNR of 34.1 dB at 0.69 bpp without entropy coding (or 0.503 bpp with an entropy coding) for their edge-based

Method	Filter	Case	Bit Rate (bpp)			PSNR (dB)	
			Level 3	Upper Bands	total		
BEEVRVQ	4-taps	Inside	0.186	0.374	0.560	33.55	
		Outside	0.186	0.373	0.559	33.00	
	6-taps	Inside	0.176	0.319	0.495	33.82	
		Outside	0.176	0.319	0.495	33.22	
	8-taps	Inside	0.177	0.311	0.488	33.75	
		Outside	0.177	0.311	0.488	33.20	
	10-taps	Inside	0.180	0.316	0.496	33.71	
		Outside	0.180	0.315	0.495	33.09	
	12-taps	Inside	0.177	0.326	0.488	33.45	
		Outside	0.177	0.322	0.488	33.01	
	16-taps	Inside	0.177	0.329	0.488	33.76	
		Outside	0.177	0.328	0.488	33.19	
	Ref. [1]	9-taps	VQ			0.78	32.10
	Ref. [26]	32-taps	VQ			0.63	32.0
	Ref. [21]		VQ			0.50	32.5
	Ref. [15]		VQ			0.69	34.1
Ref. [18]		VQ(256 × 256)			0.67	34.27	
Ref. [28]	32-taps	SQ			0.67	30.9	
Ref. [2]	16-taps	SQ			0.54	32.17	
Ref. [8]	16-taps	SQ			0.55	33.95	
Ref. [23]	16-taps	SQ			0.50	35.97	

Table 4: Coding results for "Lenna" image by BEEVRVQ and other existing schemes



Figure 10: BEEVRVQ coded "Lenna" image for inside test (left, PSNR=33.75 dB, bpp=0.488) and outside test (right, PSNR=33.20 dB, bpp=0.488), filter: 8-taps.

subband-VQ technique. Rao et al. [18] applied a multirate VQ scheme called the alphabet- and entropy-constrained vector quantization (AECVQ) to code image pyramid and reported a PSNR of 34.27 dB at 0.67 bpp for the  $256 \times 256$  "Lenna" image. Their result appears to be very good and is due, in part, to the joint exploitation of VQ and variable length entropy coding. But some practical disadvantages still remain. Among others, a substantial increase in computational complexity and memory storage requirement is a major disadvantage. In contrast, the BEEVRVQ scheme proposed here has much less design and encoding complexity due to its simple structure.

Comparing to the performance of subband decomposition combined with SQ schemes, our approach also appears to be better than those of [2, 11, 28] and is comparable to that of Gharari [8]. Recently Shapiro [23] reported a surprisingly good PSNR of 35.97 dB at 0.5 bpp by using an embedded wavelet hierarchical image coder. Although VQ can outperform SQ theoretically, such is often not the case in practice. Notice that the results obtained in [1] and [27] are not better than those of subband-SQ schemes. There are some possible reasons for this. First, the performance of VQ depends on how the statistics of the test image matches that of the training set. Notice that even when the test image is inside the training set, the performance still depends on the size of the training set, as well as the statistical similarity among different training images. Second, all the subband-SQ schemes employ entropy coding such as Huffman coding or arithmetic coding

for each subband. Some even use special strategies, e.g., a hierarchical entropy coded quantizer in [23]. Consequently they may provide good performance at very low bit rates. Also notice that [23] uses six-stage wavelet decomposition, which is more than the commonly used 2 or 3 stage decomposition. This may be one reason for its low bit rate. It is conceivable that if entropy coding is appropriately incorporated with VQ, and if the training set is good enough to reflect the characteristics of the test image, subband-VQ can very well outperform those excellent subband-SQ schemes.

Note that the bit allocation in our approach depends critically on the statistics of the training set. Thus if the training images do not have a consistent statistical character, or the statistics of the test image does not match that of the training set well, the performance may be severely affected. Some improvement can be expected if the bit allocation scheme can be modified to allocate quantizers to each subband in the best way possible for a particular image. Furthermore, note that larger threshold increases the distortion while reducing the bit rate. This is indeed a trade-off between distortion and bit rate and an optimal estimation threshold may be desirable.

It is interesting to see how the performance changes when filters of longer length, which correspond to wavelets of higher order regularity, are used. Table 4 shows that when the filter length increases from 4 to 6, a relatively large reduction in bit rate for the same PSNR is observed. However there is only small improvement or no improvement when the filter length continues to increase. This observation is similar to that made by Rioul [19] where scalar quantization is used to code the upper bands. This seems to suggest that regularity may be relevant for still image compression, at least for short filters for which the regularity order is relatively small, and that employing very regular filters is probably useless.

## 5 Subband Finite State Vector Quantization

In the last section, we tried to predict "non-edge" blocks from the coarser scale subimages. In this section, this idea will be extended to predict any kinds of blocks in the current scale subimages from the next coarser scale subimages. Because of the similarities among the subbands of "like" orientation, the pattern of a block in the current scale is likely to be similar to that of the corresponding block in the next coarser scale. In other words, the pattern of the corresponding block in the next coarser scale gives a prediction of what the shape and gain of the block in the current scale are likely to be. This property can be exploited to further reduce the bit rate. In this section we apply the concept of Finite State Vector Quantization (FSVQ) to

further exploit the relationship among the subbands of any given directional pyramid.

## 5.1 The Subband-FSVQ Algorithm

An FSVQ is a recursive VQ with only a finite number of states [6]. In FSVQ, the current state determines the codebook to be used for VQ on the input vector, i.e., it serves as a prediction of the input vector. The next state of the quantizer is determined by both the current VQ index and the current state of the quantizer.

Naveen and Woods [16] use a scalar version of FSVQ, Finite State Scalar Quantization (FSSQ), to exploit the relationship between various subbands of an image. The basic idea is similar to the one in [11]. In their subband-FSSQ algorithm, they use finite state PCM for the coding of upper bands, where the state of the quantizer at a given input sample determines the step size of the quantizer. They classify the quantizer outputs based on local variance. The classes of the outputs of the current scale subimages serve as the states for the quantizers when the samples in the next finer scale subimages are inputted to the finite state PCM.

Here we propose a subband-FSVQ (SFSVQ) algorithm to directly predict the pattern of a block in the current scale from the corresponding block in the next coarser scale. In our subband-FSVQ algorithm, the subbands are coded progressively, from coarser scales to finer scales. The quantizer output is naturally classified according to the codeword index assigned to it. The class of the current output serves as the state for the vector quantizer when the corresponding block in the next finer scale is coded, i.e., all the blocks in the next finer scale whose parent blocks are mapped to the same codeword belong to the same class, and therefore are coded using the same sub-codebook. Note that blocks belonging to the same class have the same coded parent block, therefore are very similar to one another. So relatively smaller codebook or fewer bits can be used to code them.

For the design of the state codebook, one approach is to use the blocks belonging to the same class as the training set and use a VQ design technique to construct a sub-codebook for this class. This approach, however, has some drawbacks. First, one has to construct a sub-codebook for each class. Therefore the design complexity is too much due to the large number of classes. Second, even though the sub-codebook size may be relatively small, the total number of codewords may still be very large, also due to the large number of classes. Thus the required storage is large. Third, bit allocation may be a big problem because it is not easy to decide the sub-codebook size for each class. With this consideration, we herein employ the following

approach to deal with these problems. Note that the sub-codebooks may have some overlap between each other, i.e., codewords in one sub-codebook may have very similar counterparts in other sub-codebooks. One, therefore, can exploit this overlap property to reduce the total number of codewords. Hence, instead of using different training sets to construct different sub-codebooks, one can construct a large codebook for each subband, and then generate the sub-codebook for each class out of the subband codebook by grouping all codewords which have been mapped to by at least one of the blocks belonging to this class. In this approach, one only needs to construct a few subband codebooks, thus reduces the design complexity. The total number of codewords is also reduced because we take advantage of the sub-codebook overlap property, therefore the storage requirement is small. Also the bit allocation problem becomes relatively simpler because one only needs to allocate bits among subbands rather than among classes.

At the receiver ends, the receivers store the codebook for each subband. The subbands are also decoded progressively, from coarser scales to finer scales. When a channel symbol is received, the receivers first identify which class its parent block belongs to, and then use the state label together with the channel symbol to decide which of the codewords in the subband codebook is selected. This can be accomplished using a two-stage Look-up Table (LUT) technique as shown in Figure 11. The first stage LUT outputs the indices of the codewords in the subband codebooks, with the channel symbol and the state label as inputs. The second stage LUT then outputs the desired codeword. Note that the storage requirement of the two-stage LUT is much smaller compared to the case when one constructs a sub-codebook for each class separately.

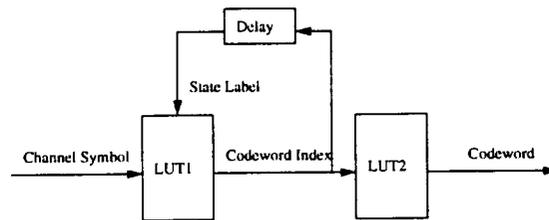


Figure 11: Two-stage LUT for the decoding of subband-FSVQ

The state quantizer design procedure and the encoding and decoding procedures are outlined in the following:

### 5.1.1. State Quantizer Design

*A:* For each directional pyramid and subband scale, design a VQ codebook according to some bit allocations based on a training set. Note that the block size increases by a factor of two along both vertical and horizontal directions when one comes from one scale to the next finer scale.

*B:* Classify the sample blocks of training set in the parent subband of the current subband into  $S$  classes, based on the codewords to which they are mapped. The classes serve as the states for the quantizers when the blocks in the current subband are coded. The corresponding blocks in the current subband are then divided into  $S$  groups, each corresponding to one state.

*C:* For each state  $s=1$  to  $S$ , group all codewords in the current subband which have been mapped to by at least one of the blocks in this state, and use it as the sub-codebook for this state. Reindex this sub-codebook. The indices will be sent out as channel symbols after the encoding phase.

### 5.1.2. Encoding Phase

After designing the quantizers as outlined above, the encoder does the following for each directional pyramid and subband scale:

*A:* Encode the lowest upper bands using the codebooks designed for these subbands.

*B:* Go to the next finer subbands. For each block in these subbands, check the class of its parent block to decide the state, and then select the corresponding sub-codebook to code this block.

*C:* Go to step *B*.

### 5.1.3. Decoding Phase

At the receiver end, decoding is performed progressively, from coarser scales to finer scales.

*A:* Decode the lowest upper bands using the codebooks designed for these subbands.

*B:* Go to the next finer subbands. For each block in these subbands, check the class of its parent block to decide the state, and then select the corresponding sub-codebook to decode this block. The sub-codebook can be found using the LUT shown in Figure 11.

*C:* Go to step *B*.

Note that in Step *A* of the state codebook design procedure, we state that the block size increases by a

factor of two along both vertical and horizontal directions when one comes from one scale to the next finer scale. In practice, however, it may not be possible because the exponentially increasing block size will incur intolerable complexity. In this case, one may break the quantized parent subband into blocks of relatively small size (say,  $2 \times 2$ ), categorize these blocks into  $S$  classes by using an extra codebook. Therefore the block size in the current subband is not too large ( $4 \times 4$  corresponding to  $2 \times 2$  in the parent subband). In this way, one can keep the block sizes in all scales reasonably large.

## 5.2 Simulation Results and Discussion

Our experiments are based on a training set that consists of seven  $512 \times 512$  black-and-white images. The  $512 \times 512$  "Lenna" image is used as the test image. For simplicity, the codebook for each upper subband is generated using a standard fixed rate GLA algorithm, and bits are allocated among the subband codebooks using the high resolution quantizer theory as in [1]. The results reported here are to demonstrate how much bit rate one can save by employing the SFSVQ scheme to further exploit the inter-band dependency.

As in Section 4, after three stages of decomposition, the lowest subband of "Lenna" image is coded by DPCM and Huffman coding. The other three subbands in Level 3 are coded by PCM plus Huffman coding. In our simulations, we have tested two cases of SFSVQ, i.e., one-stage SFSVQ and two-stage SFSVQ. In the one-stage SFSVQ, each subband in Level 2 is coded by the subband codebooks, and then is used to predict the blocks in Level 1. The block size is  $2 \times 2$  in Level 2 and  $4 \times 4$  in Level 1. In the two-stage SFSVQ, instead of directly using the subband codebooks, the coding of the  $2 \times 2$  blocks in Level 2 is based on the prediction from the parent block of size  $1 \times 1$  in Level 3. Note that we also use Predict\_4\_to\_4 technique described in Section 2 to eliminate the "non-edge" information.

The coding results are shown in Table 5. We compare the subband-FSVQ scheme with ordinary subband-VQ scheme which codes different subbands "independently", except that the "non-edge" information is also eliminated by Predict\_4\_to\_4. Therefore the gain reported in the following is purely due to the prediction strategy introduced in this section. For the one-stage SFSVQ case, we observe a reduction of about 10% in the bit rate for coding the upper bands for both inside test and outside test, comparing to ordinary subband-VQ. Equivalently, there is a reduction of about 0.5 dB in PSNR at an average bit rate of 0.6 bpp. In the two-stage SFSVQ case, however, it turns out that the additional reduction in bit rate is negligible while the constraint on the codebooks causes loss of PSNR for the outside test. This could be explained

as follows: One pixel in Level 3 can not predict the shape of the  $2 \times 2$  block in Level 2. It can not predict the gain of the  $2 \times 2$  child block well, either, due to the nonlinear downsampling operation in the subband system. Therefore, this additional stage of prediction almost saves no bits. Moreover, this stage of prediction places constraint on the coding of subbands in Level 2, hence incurs additional distortion.

The cases tested here are very simple ones. It is seen that in the case of one-stage SFSVQ, the bit rate is reduced without loss of PSNR performance. In cases that more stages of decomposition are involved, one may have more than one stages of prediction. Though more stages of prediction can reduce bit rate more if applied properly, it may also incur some loss of PSNR performance. This is due to the fact that the classification indeed puts a constraint on the sub-codebook to code an input block. This constraint makes it possible that the selected codeword may not be a very good representation for the input block, resulting in extra distortion. Worse though, the error caused by the mismatch may propagate to the next stage prediction where finer scale subbands are coded, and result in possibly very large distortion. The distortion may depend on how good the prediction is. The effects of multi-stage prediction need more investigation. One possible approach is to employ delayed decision encoding such as trellis encoding [6] to guarantee that longer sequences of input and reproduction vectors have the minimum possible distortion for the particular decoder.

Method	Case	Bit Rate (bpp)			PSNR (dB)
		Level 3	Upper Bands	total	
One-stage	Inside	0.177	0.418	0.595	35.19
SFSVQ	Outside	0.177	0.418	0.595	34.52
Two-stage	Inside	0.177	0.414	0.591	35.19
SFSVQ	Outside	0.177	0.414	0.591	33.58
SBVQ	Inside	0.177	0.470	0.647	35.19
	Outside	0.177	0.469	0.646	34.64
	Inside	0.177	0.419	0.596	34.63
	Outside	0.177	0.419	0.596	34.19

Table 5: Comparison of subband-FSVQ and ordinary subband-VQ for "Lenna" image. Filter: 8-taps.

## 6 Conclusion

This paper proposed three VQ techniques which aim to exploit both the intra-band and inter-band redundancy for coding the upper subbands. The first scheme is based on a quadtree structure in an effort to take full advantages of the intra-band and inter-band redundancy. Attempting to alleviate the implementational complexity of the first scheme, the second scheme codes each scale separately while eliminating the "non-edge" information by using a block-based edge prediction scheme. It is shown that this scheme yields better performance than other existing subband-VQ approaches. The third scheme further exploits the inter-band dependency by employing finite state VQ to predict blocks of any patterns in a scale from those in coarser scales. Simulation results show that it results in further bit rate reduction and seems to be a promising approach.

## References

- [1] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. Image Processing*, vol. 1, no. 2, pp. 205-220, April, 1992.
- [2] M. Banham and B. Sullivan, "A wavelet transform image coding technique with a quadtree structure," *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pp. 653-656, 1992.
- [3] T. Berger, *Rate-Distortion Theory: A Mathematical Basis for Data Compression*. Englewood Cliffs, NJ: Prentice-Hall, 1971.
- [4] A. Croisier, D. Esteban, and C. Galand, "Perfect channel splitting by use of interpolation, decimation and tree decomposition techniques," *Proc. of Int. Conf. on Information Sciences/System*, Patras, Greece, pp. 443-446. August 1976.
- [5] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Comm. Pure Appl., Math.* 41 (1988), pp. 909-996.
- [6] A. Gersho and R.M. Gray, *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 1992.
- [7] H. Gharavi and A. Tabatabai, "Sub-band coding of digital images using two-dimensional quadrature mirror filtering," *Proc. SPIE: Visual Communications and Image Processing*, vol. 707, pp. 51-61, September 1986.
- [8] H. Gharavi and A. Tabatabai, "Sub-band coding of monochrome and color images," *IEEE Trans. on Circuits and Systems*, vol. 35, pp. 207-214, February 1988.
- [9] S.C. Huang and Y.F. Huang, "Principal component vector quantization," *J. Visual Commun. Image Representation*, vol. 4, no. 2, pp. 112-120, March, 1993.
- [10] D.A. Huffman, "A method for the reconstruction of minimum redundancy codes," *Proc. IRE*, vol. 40, no. 9, pp. 1098-1101, Sept. 1952.
- [11] O. Johnson, O. V. Shentov, and S. K. Mitra, "A technique for the efficient coding of the upper bands in subband coding of images," *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pp. 2097-2100, 1990.
- [12] A. S. Lewis and G. Knowles, "Image compression using the 2-D wavelet transform," *IEEE Trans. Image Processing*, vol. 1, no. 2, pp. 244-250, April, 1992.
- [13] X. Li, *Diamond-Shaped Subband Image Coding*. M.S. Thesis, Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556, Nov. 1992.
- [14] S. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Trans. on Pattern Analysis and Machine Intell.*, vol. 11, no. 7, pp. 674-693, July 1989.
- [15] N. Mohsenian and N. M. Nasrabadi, "Edge-based Subband VQ Techniques for Images and Video," *IEEE Tran. on Circuits and Systems for Video Technology*, vol. 4, no. 1, pp. 53-67, February 1994.
- [16] T. Naveen and J. Woods, "Subband finite state scalar quantization," *Proc. IEEE Int. Conf. on Acous., Speech and Signal Process.*, vol. 5, pp. 613-616, 1993.
- [17] B. Ramamurthi and A. Gersho, "Classified vector quantization of images," *IEEE Trans. Commun.*, vol. COM-34, pp. 1105-1115, Nov. 1986.
- [18] R. P. Rao and W. A. Pearlman, "Multirate vector quantization of image pyramids," *Proc. IEEE Int. Conf. on Acous., Speech and Signal Process.*, vol. 4, pp. 2657-2660, 1991.
- [19] O. Rioul, "On the choice of "wavelet" filters for still image compression," *Proc. IEEE Int. Conf. on Acous., Speech and Signal Process.*, vol. 5, pp. 550-553, 1993.

- [20] E. A. Riskin and R. M. Gray, "A greedy tree growing algorithm for the design of variable rate vector quantizer," *IEEE Trans. Signal Processing*, vol. 39, no. 11, pp. 2500-2507, Nov. 1991.
- [21] R.J. Safranek, K. Mackey, N.S. Jayant, and T. Kim, "Image coding based on selective quantization of the reconstruction noise in the dominant subband," *Proc. IEEE Int. Conf. on Acous., Speech and Signal Process.*, pp. 765-768, 1988.
- [22] C.E. Shannon, "Coding theorems for a discrete source with a fidelity criterion," *IRE Nat. Conv. Rec.*, pp. 142-163, Mar. 1959.
- [23] J. M. Shapiro, "An embedded wavelet hierarchical image coder," *Proc. IEEE Int. Conf. on Acoustics, Speech, Sig. Proc.*, vol. 4, pp. 657-660, 1992.
- [24] N. Tanabe and N. Farvardin, "Subband image coding using entropy-coded quantization over noisy channels," *IEEE Journal on Selected Areas in Communications*, vol. 10, no. 5, June 1992.
- [25] P.P. Vaidyanathan, "Quadrature mirror filter banks, M-band extensions and perfect-reconstruction techniques," *IEEE ASSP Magazine*, pp. 4-20, July 1987.
- [26] P. H. Westerink, J. Biemond, and D. E. Boekee, "Sub-band coding of images using predictive vector quantization," *Proc. IEEE Int. Conf. on Acous., Speech and Signal Process.*, pp. 1378-1381, 1987.
- [27] P. H. Westerink, D. E. Boekee, J. Biemond, and J. W. Woods, "Sub-band coding of images using vector quantization," *IEEE Trans. Commun.*, vol. 36, pp. 713-719, June 1988.
- [28] J. W. Woods and S. D. O'Neil "Sub-band coding of images," *IEEE Trans. on Acous., Speech and Signal Process.*, vol. ASSP-34, no. 5, Oct. 1986.
- [29] W.J. Zeng, Y.F. Huang, and S.C. Huang, "Greedy tree growing algorithms for designing variable rate vector quantizers," *Proc. IEEE Int. Conf. on Acous., Speech and Signal Process.*, vol. 5, pp. 594-597, 1993.
- [30] W.J. Zeng, *Vector quantization for image coding*, Master thesis, Dept. of Elec. Eng., Univ. of Notre Dame, Notre Dame, IN 46556, August 1993.