Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT

Digital Signal Processing 13 (2003) 656–668

Digital Signal Processing

www.elsevier.com/locate/dsp

Quadtree-based classification in subband image coding

F. Golchin and K.K. Paliwal *

School of Microelectronic Engineering, Griffith University, Brisbane, QLD 4111, Australia

Abstract

Classification of blocks of subband samples according to their energy and variable bit allocation within the subsequent classes has demonstrated considerable gains in coding efficiency. The gains due to classification increase as smaller blocks are used; however, this increases the overheads for transmitting the classification information. In this paper, a quadtree based method is proposed. This method allows for more efficient classification by using variable-sized blocks in order to maximize the classification gain, while maintaining a limit on the classification overheads. This method is applied for subband coding of images and the results indicate that it performs better than the other methods currently available in the literature.

Keywords: Image coding; Subband; Classification; Quadtree; Trellis coded quantization

1. Introduction

Subband coding [1] is known to be an efficient method of coding images at low bit-rates. In subband coding, the image is first decomposed into a number of critically sampled subbands and then quantized and transmitted to the decoder. In a subband decomposed image, the different subbands usually contain vastly different amounts of energy. This property of subbands is utilized in coding. The bands which contain more energy are quantized using a finer quantizer and those bands which contain less energy are quantized more coarsely.

The choice of fine and coarse quantizers corresponds with the number of bits used by each quantizer. Usually an optimization algorithm [3] is used to allocate bits to each subband according to the energy in that subband and the rate-distortion characteristics

* Corresponding author.
E-mail address: k.paliwal@me.gu.edu.au (K.K. Paliwal).

doi:10.1016/S1051-2004(03)00020-4

of the quantizers being used. In effect, the optimization algorithm minimizes the mean-squared-error (MSE) of the reconstructed image for a given overall bit-rate. In this fashion the non-uniform distribution of energy across the subbands is used to achieve compression. However, a close examination of a typical subband decomposed image reveals that the spatial distribution of energy within the subbands is also far from uniform.

Most of the energy within subbands is confined to areas corresponding to edges and strong textures in the original image. This non-uniformity within the subbands can be exploited to make coding more efficient. Chen and Smith [2] proposed such a scheme for coding of images using the discrete cosine transform (DCT). In their scheme, the image is divided into a number of equal-sized square blocks which are classified according to their energy. Each DCT coefficient within each class is then assigned a number of bits according to the average energy of the particular transform coefficient in that class and the overall bit budget.

The Chen–Smith type of classification can be easily adapted for use in a subband coder. Each subband is divided into a number of equal-sized blocks which are classified according to their energies. An optimization algorithm is then used to select an appropriate quantizer for each class of each of the subbands.

In the Chen–Smith type classification, the classes are chosen such that they are equally populated. However, this is non-optimal and recently better classification schemes have been devised [8,10]. Joshi et al. [4] have provided a thorough study of a number of classification schemes for subband coding.

Although classification provides considerable coding gain, this gain comes at a cost. The decoder needs to be made aware of the classification information. This is normally done by transmitting a classification table which indicates the classes to which the blocks of subband samples belong. Using smaller sized blocks results in a higher coding gain, but it also increases the amount of classification information which needs to be transmitted.

The choice of an appropriate block size is a trade-off between the coding gain resulting from classification and the amount of classification information. Several methods for the reduction of classification information have been proposed in the literature [4]. However, at low bit-rates (less than 0.5 bpp), the classification information can still amount up to 20% of the total bit budget.

The scheme proposed in this work aims to provide more efficient classification by using smaller blocks where required (in areas of high activity) and larger blocks in other areas.

2. A classification based subband coder

As mentioned previously, the energy in the subbands is not distributed uniformly. In a typical subband decomposed image, there are small areas of high activity which correspond to edges and strong textures, and large areas with little activity corresponding to the smoother areas in the original image. We wish to exploit this property by allocating smaller block sizes over the non-uniform (high activity) areas of the subbands and larger block sizes in the areas of uniformity (low activity). This added degree of adaptivity allows for more efficient classification of the subband samples for a given classification bit budget.

In the case of non-uniform block sizes, the decoder also needs to be made aware of the sizes and the locations of the blocks used. We propose here to use the quadtree structure [9] to derive an efficient method of encoding the blocking scheme. In the following sections, we will describe the algorithms for generating the quadtrees and the methods of encoding the quadtrees.

2.1. Generating the quadtrees

Similar to the binary-tree, the quadtree is a tree structure. However, instead of nodes branching off to two children as is the case for a binary tree, the quadtree's nodes branch off to four children.

In our application, the root of the quadtree corresponds to the entire image (or a particular subband image) and each node which descends from the root corresponds to a square block within that image. The quadtrees used in this work are not balanced and hence, to encode them one bit must be sent along for each node in the tree to indicate whether or not that node is split.

Now we will examine how the quadtrees are generated. The aim of our quadtree generation process is to split an image (or subband image) into small subblocks, each of which have roughly uniform properties. The algorithm used in this case is based on growing the quadtree one step at a time, while splitting a block in each step of the growth. The choice of which block is to be split is made on the basis of an objective criterion, which depends on the degree of uniformity within the block as well as the size of the block. We will refer to this criterion as splitting gain (SG).

We define the splitting gain in two different ways. In the first definition, we use the notion of the classification gain for a non-stationary source [4] and define the splitting gain as follows:

$$\mathrm{SG} = \frac{N_p \sigma_p^2}{\prod_{i=1}^{4} (\sigma_i^2)^{1/4}}, \tag{1}$$

where N_p is the number of samples in the parent block and σ_p^2 the sample variance of the parent block. The σ_i^2's in this equation represent the variances of the four blocks which would be formed as a result of splitting the parent block. In subsequent sections, we will refer to this definition of the splitting gain as Definition A.

Looking at the problem from a slightly different perspective, we may also define the splitting gain based on how well the energy of the sub-blocks is represented by their parent. That is, if the energy (standard deviation) of all sub-blocks is similar to that of the parent, we may wish to leave that particular block unsplit. On the other hand if some subblocks have energies which differ greatly from that of the parent block, we would wish to split the block. This definition is closer in line with that of conventional quadtree based image coding. In this case, the splitting gain is defined as follows:

$$\mathrm{SG} = \frac{N_p}{4} \sum_{i=1}^{4} (\sigma_p - \sigma_i)^2. \tag{2}$$

We will refer to this definition as Definition B.

The algorithm for growing the quadtree is as follows:

1. Initialize the quadtree root to the entire image (or subband).
2. Split the root into 4 equal sized blocks.
3. Calculate the splitting gain (SG) for each block.
4. Split the block with the largest splitting gain into 4 blocks.
5. Calculate splitting gain for the new blocks.
6. Repeat from Step 4 until a maximum number of blocks is reached.

We should note that when quadtrees are utilized in a coder, the structure of the quadtree needs to be made known to the decoder. Hence, the number of bits used to encode the quadtree can be important.

Once the quadtree has been generated using Definition A or Definition B, it is simply encoded using one bit for each node in the quadtree to indicate whether or not it has been split.

Since our algorithm for generating the quadtree is based on successive splitting of the blocks, it is appropriate to evaluate the length of the quadtree in terms of the number of block splits performed:

$$\text{QuadtreeLength(bits)} = 4N_s + 1, \tag{3}$$

where N_s is the number of splits performed on the quadtree root.

In cases where a minimum block size has been set, the leaf nodes of that size will not require an additional bit since they are always left unsplit. Therefore Eq. (3) only sets an upper limit on the number of bits required to encode the quadtree.

2.2. Using the quadtrees in a subband coder

So far we have explained how quadtrees can be generated in order to split an image into blocks with uniform activity levels (as measured by variance or standard deviation). Now we will take a look at how this concept can be utilized in a classification based subband coder.

The subband coder used in this work relies on a 22-band decomposition as used in [4], which is shown in Fig. 1.

Joshi et al. [4] have experimented with a number of classification schemes. The most successful of these schemes is based on classifying equal-sized blocks (2×2 for bands 0 to 6 and 4×4 for bands 7 to 21) into one of four classes within each subband. Two algorithms named maximum classification gain and equal mean-normalized standard deviation (EMNSD) are proposed for classification. They have shown that these algorithms perform almost equally well and that they outperform the Chen–Smith [2] type of classification. Unlike Chen–Smith classification, both of these algorithms result in classes with unequal populations.

Joshi et al. [4] have also devised methods of reducing the classification information which needs to be sent along by exploiting various dependencies both between and within the classification maps of the subbands. However, despite the reductions, the classification information still comprises a large portion of the total bit-rate. With the use of the quadtree

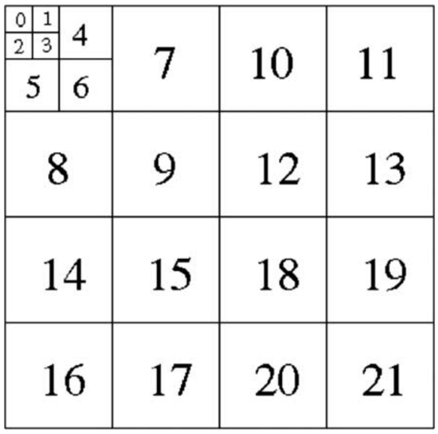

Fig. 1. 22-band subband decomposition.

structures described in the previous section, we aim to reduce this overhead through a better, adaptive choice of block sizes.

The simplest method of utilizing the quadtrees in this scheme would be to generate a quadtree for each subband and then perform the classification accordingly. However, in that case, the cost of encoding the quadtrees themselves can become prohibitive. A typical quadtree (with around 400–500 blocks) used for a subband can contribute around 0.002 bpp to the overall bit-rate for a 512×512 pixel image. Thus encoding 22 such quadtrees would take up around 0.04 bpp which is quite expensive when the total bit budget is around 0.5 bpp or less.

The smallest four subbands (subbands 0–3) require at most 512 bits to classify them into 4 classes of 2×2 blocks. This is equivalent to a contribution of approximately 0.002 bpp to the total bit-rate which is hardly worthwhile attempting to reduce. On the other hand, the higher frequency subbands 10–21 usually contain very little energy and in our range of target bit-rates are mostly quantized to zero. The subbands which interest us the most are subbands 4–9 where the majority of the classification information is required.

We examine 3 different methods for incorporating the quadtrees into the subbands classification:

- Method 1: Subbands 0–3 are divided into 2×2 sample blocks and classified. A single quadtree is generated on the original (512×512) image, and scaled down appropriately for use in subbands 4–21. The minimum block size in the quadtree is limited to 16×16 pixels which corresponds to 2×2 samples in subbands 4–6 and 4×4 samples in subbands 7–21 after appropriate down-scaling.
- Method 2: A quadtree is designed for each of the subbands 4–9. Uniform sized blocks (2×2 samples) are used for subbands 0–3. Subbands 10–21 use the same quadtree as subbands $7, 8$, or 9 depending on their orientation. That is, the diagonal bands (subbands 18 and 21) use the quadtree generated for subband 9. The vertical bands (14, 15, 16, 17, and 20) use the same quadtree as subband 8 and so on.

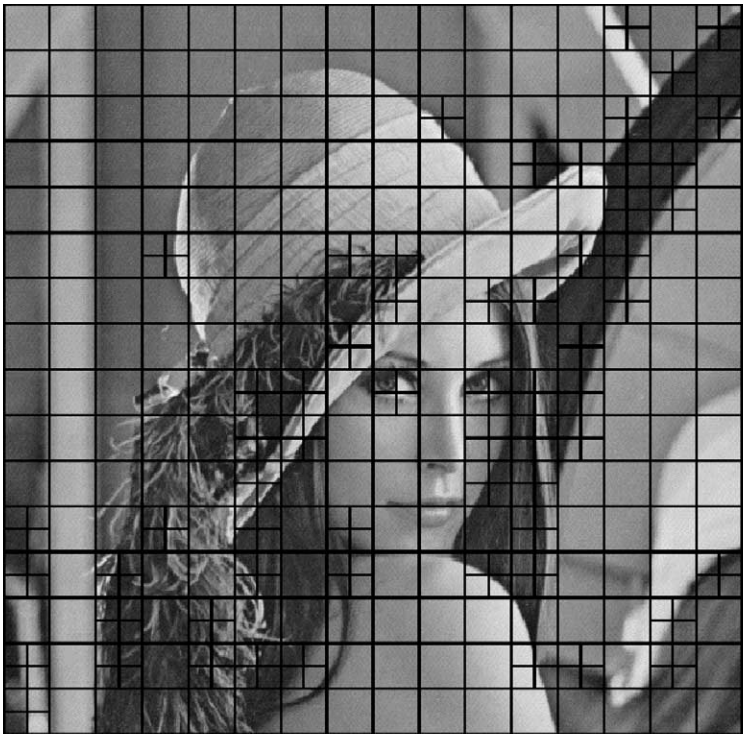

Fig. 2. Quadtree generated using Method 1A and superimposed onto the Lena image.

- Method 3: A quadtree is designed for each of the subbands 4, 5, and 6. Subbands 0–3 are divided into 2×2 blocks as before. Subbands 7–21 use scaled up versions of the quadtrees for subbands 4, 5, or 6 depending on their orientation (determined as in Method 2).

Figure 2 is an example of a quadtree (of size 421 blocks) generated from the original (512×512) Lena image by Method 1 using Definition A of splitting gain. The quadtree has been superimposed onto the original image to show where the splits have been made.

In the following sections, we provide a more detailed description of the subband coder and compare the various methods of generating and using the quadtrees.

3. The subband coder

The subband coder used to demonstrate the quadtree based classification is very similar to the coders used in [4] and [5]. The main difference between the two coders is the method of classification used in the subbands. However, in the following sections we will briefly examine the various components of the coder.

3.1. Classification within subbands

Once the subbands are divided into blocks (using quadtrees or equal-sized blocks), the blocks within each subband are classified into four classes. The classification algorithm used is the equal mean-normalized standard deviation (EMNSD) classification as described in [4].

EMNSD was initially designed for use with equal-sized blocks [8]. Unlike the Chen–Smith type of classification, it classifies blocks of samples into classes with unequal populations. The classification is based on the values of block energies and is performed so that each class contains blocks with similar statistical properties. Blocks are sorted according to their gains (sample standard deviations) and the class boundaries are chosen such that the mean-normalized standard deviations of the block gains in all classes are roughly similar.

Although the EMNSD classification scheme is based on a heuristic argument, it has demonstrated excellent performance and has outperformed Chen–Smith classification in all test cases. For a complete description of this algorithm, the reader is referred to [4].

EMNSD was selected as the classification algorithm for use with quadtrees. The unequal block sizes used in the quadtree based classification scheme do not pose a problem for the EMNSD algorithm and it can be implemented without any alteration.

3.2. Characterization of the subbands

Once the 22-band subband decomposition has been applied (using suitable filters), the samples within each higher frequency subband (HFS) are left largely uncorrelated. Hence, the samples within these bands can be efficiently treated as outputs of memory-less sources. The samples in the lowest frequency subband (LFS) will still contain some small correlation. However, due to the small size of the LFS, they can also be treated as the outputs of a memory-less source without significantly affecting the efficiency of the coder.

In the past, subbands were often modeled as realizations of either Gaussian or Laplacian sources [6]. Although this provides a suitable description for the statistics of the subband samples, more recently, researchers have turned to the use of generalized Gaussian (GG) distributions. GG distributions [7] are a family of exponential distributions which contain the Gaussian and Laplacian distributions as special cases. The use of these distributions enables more accurate modeling of the subbands. The generalized Gaussian distribution has been described in the following section.

3.2.1. The generalized Gaussian distribution

Generalized Gaussian distributions are a family of zero-mean, symmetrical distribution functions which include the Gaussian and Laplacian distributions as special cases. The probability density function (pdf) of generalized Gaussian distribution is given as

$$p(x) = \left[\frac{\nu\eta(\nu,\sigma)}{2\Gamma(1/\nu)}\right]\exp\left(-\left[\eta(\nu,\sigma)|x|\right]^{\nu}\right), \tag{4}$$

where

$$\eta(\nu,\sigma) = \sigma^{-1}\left[\frac{\Gamma(3/\nu)}{\Gamma(1/\nu)}\right]^{1/2}. \tag{5}$$

A generalized Gaussian distribution can be completely specified by its shape parameter ν and its standard deviation σ. The shape parameter ν describes the exponential rate of decay for pdf which is equal to 1.0 for the Laplacian distribution and 2.0 for the Gaussian distribution.

In order to model the classes within the subbands, we need to estimate these parameters for each of the class. The maximum-likelihood estimation [13] of these parameters is computationally very expensive. Mallat [11] has proposed several alternatives with lower computational complexity.

In [5], Joshi et al. have provided a simple, yet effective method for estimating the shape parameter ν form the mean absolute value and the standard deviation of the class samples. Furthermore, it has been shown [4] that maximum-likelihood estimation does not provide an advantage over this method (in terms of coding performance).

The value of ν is restricted to the set of values $\{0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 2.0\}$. The values 1.0 and 2.0 correspond to the Laplacian and Gaussian distributions, respectively. Expanding the set of potential values does not result in any significant gain. It was found that after the EMNSD classification, the classes almost always demonstrated Laplacian or Gaussian statistics. This tendency toward less peaked pdf's (e.g., Laplacian and Gaussian) is a direct result of classification.

3.3. The quantizer

The quantizer used is the arithmetic and trellis coded quantizer (ACTCQ) described in [5]. At the heart of the ACTCQ system, lies a scalar quantizer with uniform thresholds. The codewords of the scalar quantizer are divided into 4 subsets which are paired together to form two union subsets. Each state of the trellis is then labeled to correspond to one of the two union codebooks.

The Viterbi algorithm [16] is then used to choose the trellis path which minimizes the distance between the quantizer's inputs and its outputs. An arithmetic coder is used to encode the trellis codewords. For a more detailed description of the ACTCQ system, refer to [5].

The ACTCQ system is very similar to the uniform threshold quantizer (UTQ) used in [6]. However, the delayed decision encoding in ACTCQ provides additional gain similar to the "space packing" gain offered by vector quantizers.

ACTCQ and other similar quantizers, such as ECTCQ [12], have demonstrated excellent rate-distortion performance for the quantization of GG sources. The performance of these quantizers makes them ideal candidates for use in a subband coder. Operational rate-distortion curves for generalized Gaussian sources with different shape parameters (in this case $0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 2.0$) are generated and stored for subsequent use by the bit allocation algorithm.

The main parameter controlling the bit-rate of the ACTCQ for quantizing a particular source is the step size of the scalar quantizer used. Hence, when generating the rate-distortion curves, the step size is varied to change the bit-rate. At the same time, the step size and its corresponding bit-rate are recorded in a lookup table for subsequent use.

3.4. Bit allocation

An optimal bit-allocation algorithm is used to allocate the bit-budget among the classes in the different subbands. The bit allocation algorithm used in this work is that of Westerink et al. [3]. This algorithm is a greedy (gradient based) algorithm which traces the convex hull of all points on the rate distortion curve of the subband coder; where each point corresponds to a particular allocation of bits among the various sources.

The algorithm uses a previously known point (initially the point corresponding to 0 bit-rate) on the convex hull to find the next point on the convex hull which corresponds to a higher bit-rate. The procedure continues until the desired bit-rate has been reached.

3.5. Coding and side information

Once the bit-allocation is completed, the classification maps are encoded and transmitted to the decoder. As described in [4], a number of methods are used to reduce the classification information. These methods can be summarized into the following 3 points:

1. The classification maps of subbands where all classes are allocated zero bits, need not be transmitted.
2. If more than one class in a particular subband has been allocated zero bits then these classes can be combined together into one class.
3. The classification tables are entropy coded using conditional probabilities. The symbol probabilities are conditioned on the classification maps of other subbands as well as the class index of adjacent blocks. In this fashion, interband and intraband dependencies are exploited.

It should be noted that unlike the equal-block sized scheme, a particular block may neighbor more than one block to its immediate left. In these cases the larger neighbor is chosen as the preceding block to be used in generating the conditional probability tables. Also, if a block corresponds to an area occupied by more than one block in a previous subband, the larger of the blocks is chosen for generating the conditional probabilities.

The conditional probability tables, the shape parameters of the classes, the variance of the classes assigned non-zero bit-rates and the step size of the quantizers being used must be sent along as side information. As found in [4], it was verified that this information contributes well under 0.01 bpp to the overall bit-rate.

The decoder must also be informed of the structure of the quadtrees being used. The requirements for the encoding of the quadtrees have been outlined in Section 2.2. The next step in the coding procedure is encoding the classification maps of subbands. The conditional probability tables are used for entropy coding the classification maps of subbands which have at least one class with a non-zero bit-rate.

The final step in coding is the quantization of the subband samples. After the ACTCQ has completed the quantization of all subband samples, the encoded file sizes are measured and used to determine the bit rate of the coder.

4. Results

The subband coder described in the previous section is used to compare the performance of the various quadtree schemes described in Section 2 with the performance of the system using equal-sized blocks as in [4].

The filters used in the subband coder are Antonini et al.'s 7–9 tap perfect reconstruction filter pair [14] and Johnston's 32D (32-tap) filter [15]. For target bit-rates below 0.3 bpp, the 7–9 tap filter pair provides better performance both in perceptual and PSNR terms. The short filter lengths result in less noticeable ringing around the edges at low bit-rates. However, the 32-tap filter gives slightly better results at around 0.3 bpp and higher. Thus, the subband filters are selected in each case depending on the target bit-rate.

Coding results at a bit-rate of 0.25 bpp for the Lena image (512 × 512 pixels, 256 gray levels) are listed in Table 1. The allocation of the bit-rate between the classification and quantization has been adjusted so that the overall bit rate is as close as possible to the target bit-rate. Normally, this is not necessary; however, in this case, it is needed to enable meaningful comparisons to be made among the various methods.

The columns in Table 1 correspond to the different methods of using the quadtree (see Section 2.2), while the rows correspond to the definition of splitting gain (SG) used in the generation of the quadtree (see Section 2.1). It is clear that Definition A of the splitting gain produces better results regardless of the quadtree being used. This is not surprising, since this definition closely follows the concept of classification gain. It can also be observed that Method 3 of using the quadtree produces the best results. Method 3 is a good compromise between Method 1 (where one quadtree is generated for all subbands) and Method 2 (where a quadtree is generated for each of subbands 4, 5, 6, 7, 8, and 9).

Method 3A (quadtrees generated according to Method 3 and Definition A of the splitting gain) results in a PSNR improvement of almost 0.3 dB over the uniform block-size scheme of [4], which is a small but significant improvement. We should also note that in Method 3A the classification overhead (including the quadtrees) amounted to slightly over 0.02 bpp which is about half of that reported in [4]. This reduction in the classification information effectively leaves the quantizers with more available bits.

Figure 3 shows a comparison of coding results between Method 3A and using uniform block sizes is in [4]. It is evident that as the bit-rate increases the improvement due to the use of quadtrees becomes less significant. At a bit-rate of 0.5 bpp, the advantage of Method 3A over the uniform block sized scheme is around 0.08 dB which is almost insignificant (37.96 dB PSNR for Method 3A compared to 37.88 dB [4] for uniform block sizes). At this bit-rate other quadtree based methods demonstrate almost no advantage over the equal block-sized scheme. The classification overhead for Method 3A at 0.5 bpp is

Table 1
PSNR (dB) for various methods of encoding Lena at 0.25 bpp

	Method 1	Method 2	Method 3
SG Definition A	34.46	34.53	34.61
SG Definition B	34.31	34.38	34.44
Uniform sizes	34.32		

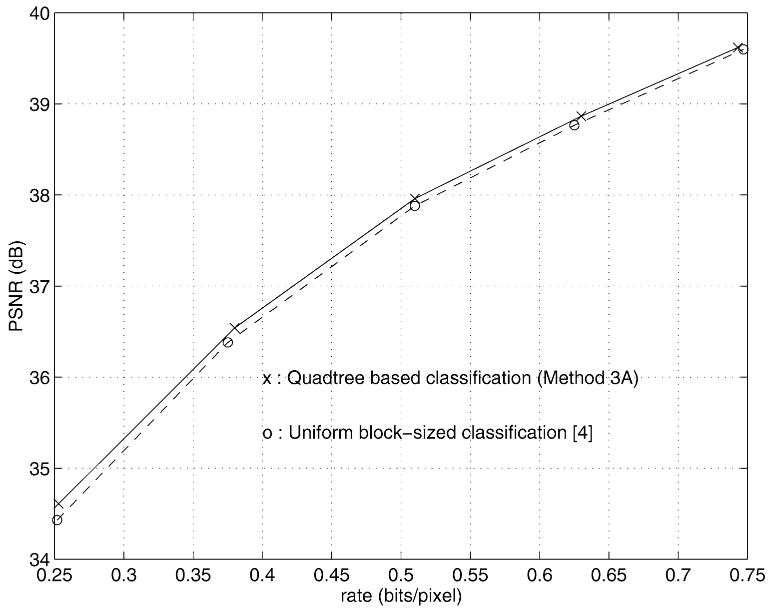

Fig. 3. PSNR comparison of Method 3A and the uniform block-sized scheme [4].

0.042 bpp which is 0.01 bpp less than the overheads for equal sized blocks. At bit-rates above 0.5 bpp the gain due to quadtrees continues to diminish as the classification information comprises an increasingly smaller portion of the total bit-rate. Table 2 provides a comparison of the PSNR performance of the Method 3A with some other recently developed coders for the Lena image.

The quadtree based scheme was also tested on the Barbara image (512×512 pixels, 256 gray levels) for comparison. The gain due to the use of quadtrees for this image is similar to that obtained for Lena. Method 3A demonstrated a gain of 0.21 dB over the uniform block sized scheme at 0.25 bpp (29.73 dB for Method 3A compared to 29.52 dB using uniform block sizes). As the bit rate increases the advantage of quadtrees becomes less significant (33.94 dB for Method 3A compared to 33.89 dB for uniform block sizes). Table 3 contains a performance comparison of Method 3A with some recent coders. It should be noted that the results in these tables are taken from the original papers. In most cases, varying the

Table 2
PSNR comparison for encoding Lena (512×512) at 0.25 bpp and 0.5 bpp

Coding method	0.25 bpp	0.5 bpp
Method 3A (quadtrees)	34.61	37.96
Subband classification [4]	34.29	37.88
Space-freq. quant. [19]	34.33	37.36
Said and Pearlman [18]	34.11	37.21
Embedded zerotrees [17]	33.17	36.28

Table 3
PSNR comparison for encoding Barbara (512 × 512) at 0.25 bpp and 0.5 bpp

Coding method	0.25 bpp	0.5 bpp
Method 3A (quadtrees)	29.73	33.89
Space-freq. quant. [19]	28.29	32.15
Said and Pearlman [18]	27.40	31.25
Embedded zerotrees [17]	26.77	30.53

filters or the decomposition scheme can result in a PSNR improvement over the results quoted in the original papers.

5. Discussion and conclusion

In this paper, we demonstrated the use of a quadtree based classification scheme for a subband coder. We have experimented with a number of different methods for the generation of quadtrees and incorporated them into a subband coder. It has been shown that Definition A of the splitting gain is an appropriate measure for use in generating the quadtree.

It has also been shown that Method 3 of using the quadtrees in the subband coder provides a good compromise between adaptivity of the quadtrees and the amount of information used in encoding individual quadtrees. The combination of Method 3 and Definition A of the splitting gain (referred to as Method 3A) results in an improvement of almost 0.3 dB PSNR over the uniform block based method of [4] at bit rates around 0.25 bpp.

This improvement is due to the reduction of the classification information which needs to be transmitted. For bit-rates below 0.3 bpp, this reduction ensures that a larger portion of the total bit budget is made available to the quantizers. As a result, improvement is quite noticeable in the quality of the coded image at these low bit-rates.

As the bit rate increases beyond 0.3 bpp, the gain due to the use of quadtrees becomes less significant. It is no longer noticeable at 0.5 bpp and beyond. This is due to the fact that the classification information comprises a smaller portion of the bit budget and, hence, has a smaller effect on the overall performance of the coder.

Other than subband classification, another group of subband coders which have demonstrated excellent results recently are coders which rely on the use of "zerotrees" [17–19]. In [19], Xiang et al. discuss the parallels between zerotrees and classification. Zerotrees can be viewed as a simple classification into two classes: zerotrees and quantized coefficients.

Xiang et al. [19] have developed an algorithm which jointly optimizes the quantization and classification (referred to as space-frequency quantization). Using a simple classification (only 2 classes) and a simple scalar quantizer, they have been able to match the results obtained using subband classification (uniform blocks) and ACTCQ at a bit-rate of 0.25 bpp.

This can be interpreted as an indication that in the uniform block-sized subband classification scheme, too large a portion of the bit-budget is being spent on classification information at low bit-rates. In quadtree based classification, this overhead is successfully reduced. The advantage of the quadtree based classification and ACTCQ over the coder proposed by Xiong et al. [19] can be attributed to the use of more than 2 classes and a better quantizer. At higher bit-rates the coder using uniform block sizes (as in [4]) considerably outperforms the coder proposed by [19]. This can be taken as an indication that the amount of classification overhead is more appropriate with respect to the target bit-rate. Again, the use of more than 2 classes (zerotree and quantized) and ACTCQ are responsible for the advantage.

References

[1] J.W. Woods, S.D. O'Neil, Subband coding of images, IEEE Trans. Acoustics, Speech Signal Process. ASSP-34 (1986) 1278–1288.
[2] W.H. Chen, C.H. Smith, Adaptive coding of monochrome and color images, IEEE Trans. Commun. COM-25 (1977) 1285–1292.
[3] P.H. Westerink, J. Biemond, D.E. Boekee, An optimal bit allocation for subband coding, in: Proc. of IEEE Internat. Conf. Acoustics, Speech and Signal Processing, April 1988, pp. 757–760.
[4] R.L. Joshi, H. Jafarkhani, J.H. Kasner, T.R. Fischer, N. Farvardin, M.W. Marcellin, R.H. Bamberger, Comparison of different methods of classification in subband coding of images, IEEE Trans. Image Process. IP-6 (1997) 1473–1487.
[5] R.L. Joshi, V.J. Crump, T.R. Fischer, Image subband coding using arithmetic coded trellis coded quantization, IEEE Trans. Circuits and Systems Video Technol. 6 (1995) 515–523.
[6] N. Tanabe, N. Farvardin, Subband image coding using entropy-coded quantization over noisy channels, J. Select. Areas Commun. 10 (1992) 926–943.
[7] N. Farvardin, J.W. Modestino, Optimum quantizer performance for a class of non-Gaussian memoryless sources, IEEE Trans. Inform. Theory IT-30 (1984) 485–497.
[8] H. Jafarkhani, N. Farvardin, C.-C. Lee, Adaptive image coding based on the discrete wavelet transform, in: Proc. of IEEE Internat. Conf. Image Processing, Vol. 3, November 1994, pp. 343–347.
[9] J. Vaisey, A. Gersho, Image compression with variable block size segmentation, IEEE Trans. Signal Process. 40 (1992) 2040–2060.
[10] R.L. Joshi, T.R. Fischer, R.H. Bamberger, Optimum classification in subband coding of images, in: Proc. of IEEE Internat. Conf. Image Processing, Vol. 2, November 1994, pp. 883–887.
[11] S.G. Mallat, A theory for multi resolution signal decomposition: The wavelet representation, IEEE Trans. Pattern Anal. and Machine Intell. 11 (7) (1989) 674–693.
[12] M.W. Marcellin, On entropy constrained trellis coded quantization, IEEE Trans. Commun. (1994) 14–16.
[13] F. Muller, Distribution shape of two-dimensional DCT coefficients of natural images, Electron. Lett. 29 (1993) 1935–1936.
[14] M. Antonini, M. Barlaud, P. Mathieu, I. Daubechies, Image coding using the wavelet transform, IEEE Trans. Image Process. IP-1 (1992) 205–220.
[15] J.D. Johnston, A filter family designed for use in quadrature mirror filter banks, in: Proc. of IEEE Internat. Conf. Acoustics, Speech and Signal Processing, April 1980, pp. 291–294.
[16] G.D. Forney Jr., The Viterbi algorithm, Proc. IEEE 61 (1973) 268–278.
[17] J.M. Shapiro, Embedded image coding using zerotrees of wavelet coefficients, IEEE Trans. Signal Process. Special Issue Wavelets and Signal Process. 41 (1993) 3445–3463.
[18] A. Said, W.A. Pearlman, A new, fast and efficient image codec based on set partitioning in hierarchical trees, IEEE Trans. Circuits and Systems Video Technol. 5 (1996) 243–250.
[19] Z. Xiang, K. Ramchandran, M.T. Orchard, Space-frequency quantization for wavelet image coding, IEEE Trans. Image Process. 6 (1997) 677–693.