ARITHMETIC CODED VECTOR SPIHT WITH CLASSIFIED TREE-MULTISTAGE VQ FOR COLOR IMAGE CODING*

Debargha Mukherjee and Sanjit K. Mitra
Image Processing Laboratory
Department of Electrical & Computer Engineering, University of California,
Santa Barbara, CA 93106.
Email: (debu,mitra)@iplab.ece.ucsb.edu,
Tel: (805) 893-8312, FAX: (805) 893-3262.

Abstract - A vector extension of the Set Partitioning in Hierarchical Trees (SPIHT) algorithm, named vector-SPIHT (VSPIHT), using trained classified successive refinement VQ, has recently been proposed. In this work, vector set-partitioning is applied to multispectral image compression, in particular to 24-bit color images. Since the individual spectral components are sufficiently correlated, VSPIHT can effectively exploit both the inter-component redundancy as well as the spatial redundancy within each subband of each component, to yield performance superior to separate scalar SPIHT coding of each component. Adaptive arithmetic coding of the first stage VQ index for each class, as well as the significance information, further improves the performance. Coding results demonstrate that the vector-based approach for color images significantly outperforms the scalar counterpart in the mean-squared-error sense.

1. INTRODUCTION

The wavelet transform [1]-[10] has grown to be very effective for compression of natural images. While the efficiency of a wavelet based image compression scheme depends both on the wavelet filters chosen, as well as on the coefficient quantization scheme employed, in this work, we focus only on the latter. Following the introduction of the zerotree prediction methodology by Shapiro in EZW [2], Said and Pearlman [3] developed an improved scheme, called set partitioning in hierarchical trees (SPIHT). Both the schemes rely on partial ordering of the wavelet coefficients by magnitude, followed by bit plane by bit plane progressive refinement. The bitstream generated is perfectly embedded. Several enhancements, such as [4], has been proposed to these algorithms. In our previous work [5], [6], we grouped several wavelet coefficients into vectors, and quantized them in the set partitioning framework, using trained, classified, successive refinement vector quantization [11]. A lattice VQ variation of the same has been developed [7]. Earlier, da Silva et al. [8] grouped wavelet coefficients into vectors in the EZW framework, and quantized them using a lattice based gain-shape type VQ. Recently Knipe et al. [9] extended the scheme to SPIHT [3], using modified shape codebooks based on the Λ_{16} lattice, which yielded superior performance.

In this work, we adopt the trained vector set-partitioning approach to code multispectral images, in particular, 24-bit color images. For most multispectral images, the individual spectral components are significantly correlated. Additionally, the neighboring coefficients within each subband of each component are correlated. Trained VQ can exploit both these correlations jointly, if the vectors are formed by

* This work was supported by ONR grant N00014-95-1-1214.

combining coefficients across the spectral components. Further, adaptive arithmetic coding of the VQ indices can be used to exploit repetitive patterns that typically occur in images. The set-partitioning framework preserves the embedding property, thereby making the scheme applicable to requirements like progressive transmission. Separate scalar or vector set-partitioning of each spectral component would not be efficient because the inter-component correlations are not effectively exploited in quantization, and the significance information is unnecessarily duplicated. Also, the across component vectoring approach simplifies the problem of bit-allocation between the individual spectral components. It is to be noted, that this independently developed work is similar in concept to the recent work of Amato et al. [10] for multispectral image compression, but is significantly different in the way successive refinement is performed with classified VQ. Some of the enhancements in this work are equally applicable to gray image compression.

In Section 2 the coding algorithm is explained in detail. In Section 3, we present the implementation details and the coding results in comparison with the scalar SPIHT scheme. Section 4 concludes the paper.

2. VSPIHT FOR MULTISPECTRAL IMAGES

2.1 Coding Scheme

The essence of color image compression using vector set-partitioning is to form vectors by combining wavelet coefficients across the various spectral components, rather than perform a scalar quantization of the individual components. Let us assume that a class of multispectral images to be coded has K components. First, a dyadic wavelet decomposition of each spectral component is performed. Then, the wavelet transform coefficients in each $H \times V$ window in each subband of each spectral component are grouped with the corresponding coefficients from the other components to form a single vector of dimension HVK. The parent child relationship between the vectors in different subbands is defined as for scalars in [3], and is shown in Figure 1. Here, each small square represents a HVK-dimensional vector of wavelet coefficients.

In order to bring about a certain amount of uniformity in the way images with varying range of wavelet coefficients are coded, the wavelet vectors as formed above are each scaled by a factor γ, given by:

$$g = \frac{R_{-1}}{max(\|v\|)} \tag{1}$$

where R_{-1} is a pre-defined parameter, and $max(\|v\|)$ denotes the maximum vector magnitude in an image. The factor γ is transmitted to the decoder with high precision for reconstruction. After scaling, all vectors are guaranteed to lie within a HVK-dimensional shell of radius R_{-1}.

The set-partitioning methodology, with three ordered lists, is now used to classify the vectors by vector-magnitude. Vector set-partitioning operates in multiple passes, where each pass is associated with a vector magnitude threshold. Each new pass yields a new class of vectors which have magnitudes higher than the threshold associated with the pass, but lower than that associated with the previous pass. The threshold progressively decreases from one pass to the next. In other words, each pass ascertains as significant the set of vectors that lie within a HVK-dimensional shell, bounded on the inside by a hypersphere of radius equal to the current threshold, and on the outside by a hypersphere of radius equal to the previous threshold. Figure 2 shows the decreasing magnitude thresholds R_0, R_1, R_2,..., with $R_{-1} > R_0 > R_1 > R_2 >$..., and the corresponding classification in HVK-dimensional space. Note that the first class, $Class_0$, is bounded on the outside by R_{-1}.

The use of the L_2-norm (magnitude) in determining significance of a vector in

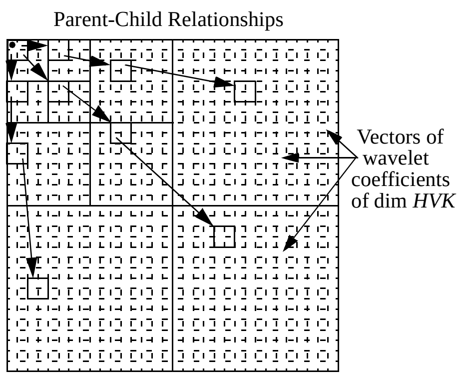


Figure 1. Parent-child relationships between vectors

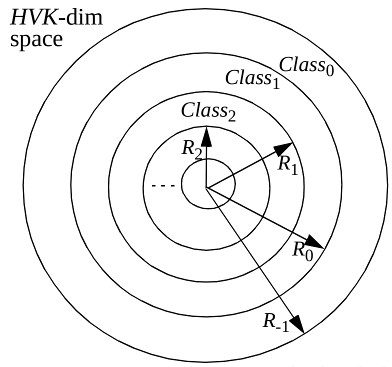


Figure 2. Decreasing magnitude thresholds to determine significance of vectors, and the corresponding classes.

a pass is justified for orthogonal wavelets, because it follows from Parseval's relationship that the squared magnitude error in quantization of the vectors contribute additively to the overall reconstruction squared-error for all the components. That is, a higher magnitude vector when transmitted losslessly, will reduce the reconstruction mean-squared error more than a lower magnitude vector, and therefore should be quantized before the other. Although this is not strictly true for bi-orthogonal wavelets, under the assumption that bi-orthogonal wavelets are approximately orthogonal, the L_2-norm is still a good criterion to use to classify the vectors. Note that the bit-allocation problem across the various spectral components no longer exists, because the algorithm automatically chooses the optimal allocation in the overall squared error sense.

2.2 Successive Refinement Classified VQ

The vectors decided as significant in a pass, are roughly quantized in the same pass, and are successively refined in the subsequent passes. The pass in which a vector becomes significant also classifies the vector, and determines a particular successive refinement VQ system to use to quantize it. The codevectors of the VQs span the shell between two concentric hyperspheres. It is necessary to have as many VQ systems as are classes, in order to exploit the class distribution pattern effectively. We observe that at the higher scales the coefficients forming a vector are very similar, and as such, the distribution of vectors in the shell is biased along the {1,1,1,...} axis. At the lower scales, on the contrary, only one or two high magnitude coefficients contribute to the significance of a vector. As such the distribution of vectors in the shell is biased along the {1,0,0,...} axis or one of its permutations. Therefore, if *N* passes are used in all, *N* successive refinement VQs are designed, one for each class.

Although a full tree-structured VQ (TSVQ) [11] is the most efficient successive refinement system, its storage requirements are enormous. To make a compromise between storage complexity and efficiency, the first stage VQ for each class is tree-structured, and all successive refinements are made by multistage VQs [11]. The multistage VQs use a staggered bit allocation rather than a uniform bit-allocation from pass to pass. That is, the significant vectors are refined in alternate passes rather than in all passes uniformly. This is done because of the fact that a 2 stage VQ is always less efficient than a single stage VQ. For example, if 6 bits are to be used for refinement of a vector in 2 passes, it is more efficient to design a single stage VQ with (6,0) bit allocation than to design a 2-stage VQ with (3,3), (4,2) or (5,1) bit allocation.

2.3 Adaptive Arithmetic Coding

To enhance the rate distortion performance of the VSPIHT coder, two different kinds of adaptive arithmetic coding is performed. The first is aimed at exploiting repetitive patterns in images. When patterns or colors (for color images) repeat in an image, similar wavelet vectors recur within the same subband in the same pass. Similar vectors, when vector quantized coarsely using the first stage VQ, are likely to yield the same encoding index. This implies that the first stage VQ index is likely to repeat if repetitive patterns occur. Adaptive arithmetic coding of the first stage VQ index for each class and each subband is used to exploit this redundancy. The adaptive arithmetic coder progressively assigns smaller and smaller codelengths to repeating indices. In order to allow the models to adapt fast enough to the underlying statistics, it is necessary that the first stage VQ, which is also tree-structured, be designed with relatively few codevectors. Our experiments show that the gains are more impressive for color images with across component vectoring than for gray images with intra band vectoring.

The second kind of adaptive arithmetic coding is aimed at reducing the significance information bits, in a manner similar to scalar SPIHT [3]. The vectors in the lists are maintained in groups of 2×2, and the significance information for the group is transmitted jointly using multiple adaptive context models.

3. IMPLEMENTATION AND RESULTS

The above coding scheme is applied to the compression of 24-bit RGB color images. The scheme is equally applicable to other color representations, including perceptually uniform ones. A 5-stage dyadic wavelet decomposition of each color component is made using the 7/9 bi-orthogonal wavelets given in [1].

The simplest VSPIHT scheme just groups the corresponding red, green and blue coefficients into vectors of dimension 3. Nine successive refinement systems for 9 corresponding classes are designed. The first stage VQ for each class is balanced tree-structured, while the rest of the stages form a multistage VQ. The thresholds and the bit-allocations for the tree-multistage class VQs are shown in Table 1. The numbers in the third column separated by commas denote the number of bits used in successive passes starting from the pass in which a class becomes significant. Note the staggered bit-allocation used for multistage VQ. A set of about 150 images of various sizes are used as the training set to design the VQs. Each original sample vector is used to generate 12 training vectors by taking all permutations of itself, and its negative vector as well. This VSPIHT scheme with dimension 3 vectors is tested on the 512×512 *Peppers* image, which was excluded from the training set used to design the successive refinement VQs. Table 2 compares for the *Peppers* image, the overall PSNR results obtained by arithmetic coded VSPIHT, against that obtained by arithmetic coded scalar SPIHT of the red, green and blue components separately, with equal bit allocation. The overall PSNR is calculated from the combined mean-squared-error of the red, green, and blue components.

A second VSPIHT scheme groups the red, green and blue components in each 2×1 window in each subband to form vectors of dimension 6. Table 3 shows the thresholds and the bit-allocations for the tree-multistage class VQs designed with permutations of data taken from a training set of 150 images. The scheme is tested on the 512×512 *Baboon* image, which was not included in the training set. The PSNR result comparisons are presented in Table 4.

We now compare the visual quality of the coded images, *Peppers* and *Baboon*, against the originals. Figure 3 shows the easy-to-code original *Peppers* image and that obtained by reconstruction at 0.3 bpp, a compression ratio of 80:1. Figure 4 shows the difficult-to-code original *Baboon* image and the one obtained by reconstruction at 1.2 bpp, a compression ratio of 20:1.

4. CONCLUSION

A low bitrate embedded color image compression scheme based on the wavelet VSPIHT methodology and adaptive arithmetic coding has been developed. Our coding results vastly outperform the scalar SPIHT algorithm when used for color image coding. Future work will involve designing higher dimensional classified VQ systems over larger windows, where the inter component redundancy, as well as the intraband redundancy within a component, can be more effectively exploited.

5. REFERENCES

[1] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, “Image coding using wavelet transform,” *IEEE Trans. Image Processing*, vol. 1, pp. 205-220, April 1992.

[2] J. M. Shapiro, “Embedded image coding using zerotrees of wavelet coefficients,” *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3445-62, Dec. 1993.

[3] A. Said and W. A. Pearlman, “A new, fast, and efficient image codec based on set partitioning in hierarchical trees,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243-50, June 1996.

Table 1. Bit Allocation for various classes with vectors of dimension 3.

Class i	Threshold R_i (R_{-1}=6144)	TS-MS-VQ Bit Allocation
0	2048	4,8,0,7,0,6,0,6,0
1	1024	4,7,0,6,0,6,0,6
2	512	5,6,0,5,0,5,0
3	256	5,6,0,5,0,5,
4	128	5,5,0,5,0
5	64	4,5,0,5
6	32	4,5,0
7	16	4,4
8	8	4

Table 2. Overall PSNR (dB) vs. Bitrate (BPP) results for the *Peppers* image

BPP	SPIHT	VSPIHT
0.15	26.05	27.18
0.3	29.80	31.22
0.45	31.90	33.23
0.6	33.46	34.68
0.75	34.52	35.79
0.9	35.40	36.56
1.05	36.18	37.23
1.2	36.80	37.96
1.35	37.30	38.53
1.5	37.77	38.94

Table 3. Bit Allocation for various classes with vectors of dimension 6.

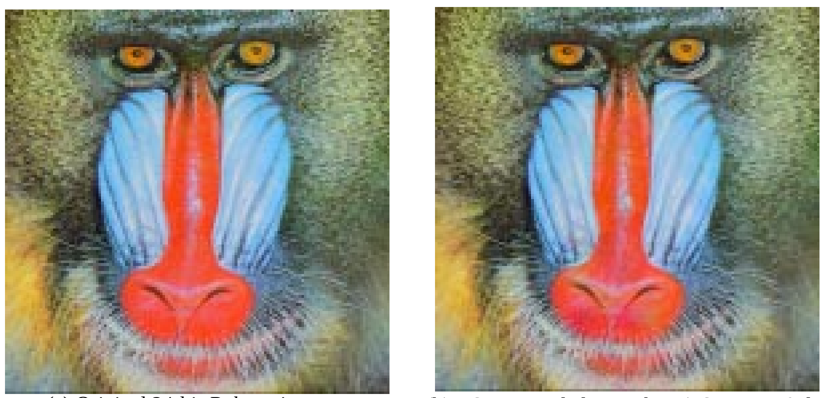
Class i	Threshold R_i (R_{-1}=6144)	TS-MS-VQ Bit Allocation
0	2048	5,9,6,9,0,9,0,9,0
1	1024	5,9,6,9,0,9,0,9
2	512	5,10,3,9,0,9,0
3	256	5,10,2,9,0,8
4	128	6,8,0,8,0
5	64	5,7,0,7
6	32	5,5,0
7	16	4,4
8	8	4

Table 4. Overall PSNR (dB) vs. Bitrate (BPP) results for the *Baboon* image

BPP	SPIHT	VSPIHT
0.3	20.84	21.53
0.6	22.07	22.98
0.9	23.09	24.45
1.2	23.94	25.20
1.5	24.71	25.85
1.8	25.53	26.44
2.1	26.25	27.00
2.4	26.87	27.66
2.7	27.44	28.19
3.0	28.03	28.48

(a) Original 24-bit *Peppers* image (b) VSPIHT coded at 0.3 bpp (PSNR 31.22 dB)

Figure 3. Visual Quality results for the Peppers Image.



(a) Original 24-bit *Baboon* image (b) VSPIHT coded at 1.2 bpp (PSNR 25.20 dB)

Figure 4. Visual Quality results for the Baboon Image.

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

[5] D. Mukherjee, and S. K. Mitra, "Vector set partitioning with classified successive refinement VQ for embedded wavelet image coding," *Proc. International Symposium on Circuits and Systems*, Monterey, California, vol. 4, pp. 25-28, June 1998.

[6] D. Mukherjee, and S. K. Mitra, "Vector set partitioning with classified successive refinement VQ for embedded wavelet image and video coding," *Proc. International Conference on Acoustics, Speech, and Signal Processing*, Seattle, Washington, vol. 5, pp. 2809-12, May 1998.

[7] D. Mukherjee, and S. K. Mitra, "Vector set partitioning with successive refinement Voronoi lattice VQ for embedded wavelet image coding," *Proc. International Conference on Image Processing*, Chicago, Illinois, Oct 1998 - to be published.

[8] E. A. B. Da Silva, D. G. Sampson, M. Ghanbari, "A successive approximation vector quantizer for wavelet transform image coding," *IEEE Trans. Image Processing*, vol. 5, no. 2, pp. 299-310, Feb. 1996.

[9] J. Knipe, X. Li, and B. Han, "An improved lattice vector quantization scheme for wavelet compression," *IEEE Trans. Signal Processing*, vol. 46, no. 1, pp. 239-43, Jan 1998.

[10] F. Amato, C. Galdi, G. Poggi, "Embedded zerotree coding of multispectral images," *Proc. International Conference on Image Processing*, Santa Barbara, California, vol.1, pp. 612-15, Oct 1997.

[11] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Boston, MA: Kluwer, 1992.