Vector Quantization with Zerotree Significance Map for Wavelet Image Coding

Sharon M. Perlmutter Keren O. Perlmutter Pamela C. Cosman†

Information Systems Lab, EE Dept., Stanford University, Stanford, CA 94305-4055
†ECE Dept., University of California, San Diego, La Jolla, CA 92093-0407

Abstract

Variable-rate tree-structured vector quantization is applied to the coefficients obtained from an orthogonal wavelet decomposition. The set of vectors from different levels of the decomposition that correspond to the same orientation and spatial location are examined in various "zerotree" groups to determine the different bit rates and distortions achievable for the set. The decision not to code certain groups of vectors is based upon choosing the desired distortion/rate tradeoff from among the possibilities. Side information is sent to the decoder to inform it of the sequence of decisions. The resulting bit stream is entropy coded. Results of this method on the test image "Lena" yielded a PSNR of 30.16 dB at 0.148 bpp.

1 Introduction

Wavelet encoding involves taking the discrete-time wavelet transform of an image and quantizing the wavelet coefficients based on some bit allocation scheme. *Bit allocation* is the process of assigning a given number of bits to a set of different sources (e.g. wavelet subbands) to minimize the overall distortion of a coder. A simple and effective wavelet coding method has been the embedded zerotree wavelet (EZW) algorithm [4]. The basic premise of the algorithm is to take advantage both of the many zeros that appear in the quantized coefficients and of the location correspondence between those zeros. This location correspondence is described as follows. In an octave-band decomposition, each coefficient x (except those in the DC band and the three highest subbands) has four children coefficients in the next higher subband. These four coefficients correspond to the same orientation and spatial location as x does in the original image. Each of these four children has four children in the next higher band. This relationship continues until the highest subband. The coefficients in the three highest subbands have no children. All of these children coefficients are referred to collectively as the *descendants* of x. A wavelet coefficient x is called *insignificant* with respect to a threshold T if $|x| < T$. The EZW design is based on the hypothesis that if a wavelet coefficient at a coarse scale is insignificant with respect to a given threshold T, then the descendants of that coefficient are likely to be insignificant with respect to T also. A coefficient is said to be an element of a *zerotree* for threshold T if itself and all of its descendants are insignificant with respect to T. An element of a zerotree is a *zerotree root* if its parent is not an element of a zerotree. When wavelet coefficients are to be coded at low bit rates by scalar quantization followed by entropy coding, the zero symbol will be the most probable symbol after quantization, and a binary *significance map* can be used to indicate the location of the non-zero values. The EZW algorithm uses the zerotree structure as an efficient way to convey significance map information. An extension to this algorithm that attempted to jointly optimize the scalar quantizers and the zerotree structures by examining distortion/rate tradeoffs was considered in [5].

In this paper, we quantize wavelet coefficients by combining variable-rate tree-structured vector quantizers (TSVQs) with a zerotree significance map that indicates the location of those coefficients that will be coded. A binary TSVQ consists of a tree with nodes labeled by candidate reproduction vectors [2]. The encoder performs a sequential binary search for the nearest neighbor until a terminal node is reached. The label of the terminal node is the final reproduction, and the binary vector describing the sequence of encoder decisions is the index that is sent to the decoder. The decoder then performs a table lookup to produce a local reproduction. A variable rate code can be implemented by an unbalanced tree, obtained either by growing a balanced tree and then pruning it back so that it becomes unbalanced, or by "greedily" growing an unbalanced tree directly [2]. A simplified version of the zerotree significance map was used with TSVQ in [1]. In that work, after a vector was encoded, the vector's descendants were examined to see if they were all insignificant with respect to some predetermined threshold. One bit of side information was transmitted to the decoder to inform it of the decision. When the encoder reached those later subbands, those vectors previously marked as insignificant would not be coded. Preliminary results were also mentioned on a method where the decision to zero out future vectors is based on a distortion/rate tradeoff, rather than a strict thresholding criterion. The current paper improves those results by extending this method in several ways. We achieve approximately a 15% reduction in bit rate by allowing the zerotree to root lower down in the hierarchy, changing the vector-forming strategy, modifying the training sequence to reflect the distribution of vectors under the zerotree method, and entropy coding the output bit stream (a 12% reduction is obtained without the entropy coding). Additional improvement in performance is achieved at the higher bit rates when a multistage version of the algorithm is implemented.

2 Zerotree Choice Algorithm

In the current work, the Daubechies orthogonal 8-tap filter was used to decompose the images 4 levels. This decomposition produced 13 subbands. Different vector sizes and shapes were used in the different subbands. The lowest band was coded by scalar quantization, the finest scales were encoded using 4×4 vectors, and intermediates scales employed vectors of size 2, 4, or 8. Figure 1 shows the decomposition together with the size and shape of the vector used in each subband. Separate TSVQs were greedily grown and optimally pruned from the training sequence in each band. For each band, the distortions and rates for the sequence of pruned subtrees provided distortion/rate curves. The minimum overall distortion for a given total rate is achieved by choosing the point of equal slope along the $D(R)$ curve for each band [3]. This step is equivalent to selecting the point where

$$\frac{\partial D_i(R_i)}{\partial R_i} = -\lambda \tag{1}$$

for all subbands i.

Since we use vectors of varying dimension at each subband level, the relationship between elements to be coded at each level is slightly modified from the scalar case. For example, a 2-D vector x in subband 2 is in spatial correspondence with two 4-D vectors of subband 5. Each of these 4-D vectors is in spatial correspondence with two 8-D vectors of subband

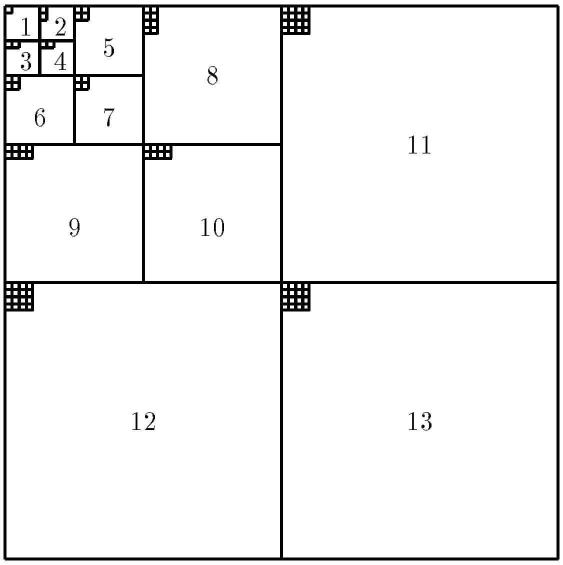

Figure 1: Sizes and shapes of the vectors chosen for each subband.

8, and each of these 8-D vectors is in spatial correspondence with two 16-D vectors of subband 11. This group consists of 15 vectors or 170 coefficients. Figure 2 illustrates an example of this group. The labels in the figure indicate distinct vectors, where the dimension of each vector is as shown in Figure 1. We will investigate coding groups of this type in order to determine the location of zerotrees. There are a number of ways to code this collection of 170 coefficients. The entire group could be declared a zerotree, without consideration of what threshold level would actually make them a zerotree. This would lead to the possible operating point (R_0, D_0). Alternatively, the lone ancestor (in subband 2) could be encoded with the TSVQ for that band, and the decoder could be told that its descendants constitute a zerotree, again without consideration of what threshold value would be required for them all to be insignificant. This would lead to a different distortion D_1 and rate R_1 associated with the entire collection of 170 coefficients. If the descendants of only one of the children vectors in subband 5 were declared a zerotree, but not the descendants of the other, this would lead to additional possibilities for D and R. Note that the rates for each of these choices consist of the sum of the rates to code the vectors that are not zeroed out and the rate to inform the decoder which of the choices to select. In all, we will consider 27 ways that the group can be encoded. These options can be illustrated with a tree structure, as shown in

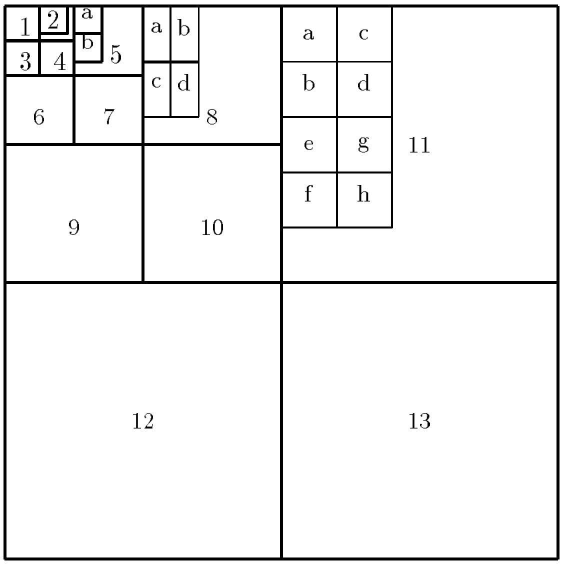

Figure 2: Group of 15 vectors considered when assigning zerotrees

Figure 3.

There are several techniques that can be used to determine the number of bits needed to inform the decoder which vectors are to be coded. The simplest approach is to use the tree depicted in Figure 3 as the bit assignment strategy. This tree is derived from making the decision of whether to zero out the vectors from the lowest subbands to the highest subbands. For each yes/no decision in the tree, a single bit would be used to inform the decoder of the result. This is the method which was implemented for this paper. If additional complexity can be tolerated, however, the bit assignment strategy can be designed using a lossless technique such as a Huffman or adaptive arithmetic coder.

Each of the 27 choices produces a (D, R) pair. These pairs can be plotted in the D, R plane and the convex hull of the set can be extracted. This curve, which we call the zerotree choice curve, then represents the distortion/rate tradeoffs obtainable for the group of 170 coefficients by different choices of where to root the zerotrees. Recall that the overall rate for each subband was chosen by finding points along the $D(R)$ curves for each subband that had equal slope. The same slope can be used to guide the selection of the operating point along the zerotree choice curve for each group of 170 coefficients. In this way, groups of coefficients are zeroed out because doing so makes sense according to a distortion/rate tradeoff.

When the training sequence in each subband consists of *all* of the training vectors that correspond to that band, the training sequence is not quite representative of the test vectors to be encoded later. This is because when the zerotrees are used, fewer test vectors of low magnitude are encoded. The training sequence can be made more representative of the test sequence by iterating the code design procedure. That is, after using all training vectors initially to design the TSVQs, the training images can themselves be encoded, and those training vectors which are declared zerotrees can be left out of the training sequence for the next round of TSVQ design. Note that this technique is also a form of classified VQ. The classifier is the zerotree structure, and it divides the training sequence into two classes. The codebook of one class will consist of one codeword with value zero, and the codebook of the second class will be constructed from only those training vectors that mapped into it (the vectors that were declared significant). This process can be iterated.

The zerotree choice algorithm can be improved if it is combined with multistage TSVQ. Multistage TSVQ (also referred to as residual TSVQ), has proven to be an effective technique for applications that possess storage or search complexity constraints [2]. In multistage TSVQ, the computational complexity increases linearly with the dimension-rate product. In addition, an important advantage to multistage TSVQ systems is that it can ameliorate training sequence size problems.

The codebook corresponding to a multistage TSVQ can be viewed as the collection of all possible reproductions that can be constructed by adding one codeword from each of the stages. The first stage encodes all of the input vectors. Then a second stage quantizer operates on the error vector between the original vector and the quantized first stage output. The quantized error vector provides a refinement to the first approximation. At the decoder, the reproduction vectors produced by the first and second stages are added together. Additional stages of quantizers can be used to provide further refinements.

In this paper a 2-stage zerotree choice system was implemented. Separate Huffman codes were designed for each band in each stage. The Huffman codes were designed non-adaptively.

3 Results and Conclusions

The training sequence was composed of 10 images from the USC database. Each image was transformed and the resulting subbands were blocked into vectors (according to the dimensions shown in Figure 1). The image "Lena" was used as a test image, and was not part of the training sequence. Each vector of the transformed test image was encoded by the pruned subtree for its subband. One iteration of training sequence reduction was implemented. The performance of the algorithm was evaluated using PSNR $= 10 \times log_{10}(\frac{255^2}{distortion})$. Figure 4 illustrates the PSNR performance of the multistage TSVQ/zerotree choice method with entropy coding, the TSVQ/zerotree choice method with entropy coding, the TSVQ/zerotree choice method without entropy coding, the TSVQ/zerotree with threshold method presented in [1], and a wavelet/TSVQ system that does not examine the significance of later coefficients (i.e., no zerotree coding). The graph demonstrates that the TSVQ/zerotree choice method performed about 1 dB better at the lower rates than the TSVQ/zerotree with threshold method, and over 1.5 dB better than the method with no zerotree coding. The multistage system performed comparably with the 1-stage system at low bit rates, and provided over 0.8 dB improvement at the higher rates. As an example, the test image had a PSNR of about 30.16 dB when it was encoded to the bit rates of 0.148 using the TSVQ/zerotree choice method (1 or 2 stages), 0.154 bpp with the TSVQ/zerotree choice system without entropy coding, and 0.27 bpp with the system that did not use any zerotree coding. We note that for the zerotree choice method the zerotree bits comprised about 8% of the total bit rate (at most bit rates).

The original image is shown in Figure 5, and the encoded image using the TSVQ/zerotree choice method with entropy coding is shown in Figure 6. Figure 7 demonstrates the significance map for the Lena image at 0.148 bpp. The white areas indicate those coefficients that were coded, the gray areas indicate those coefficients that were zerotree roots, and the black areas indicate those coefficients that were descendants of a zerotree root.

We note that VQ applied to wavelet coefficients has been shown by many researchers to produce excellent quality images at low bit rates. Some of those methods use the vector quantizer (e.g., a lattice or a classified VQ) in a purely intraband and memoryless fashion. Our method of using a distortion/rate tradeoff to declare vectorial zerotrees may be useful in such systems.

Acknowledgments

This work was funded by an AT&T Fellowship, the National Science Foundation under Grant NSF MIP-9016974, NSF Postdoctoral Fellowship, and the Army Medical Research and Material Command under Grant DAMD17-94-J-4354.

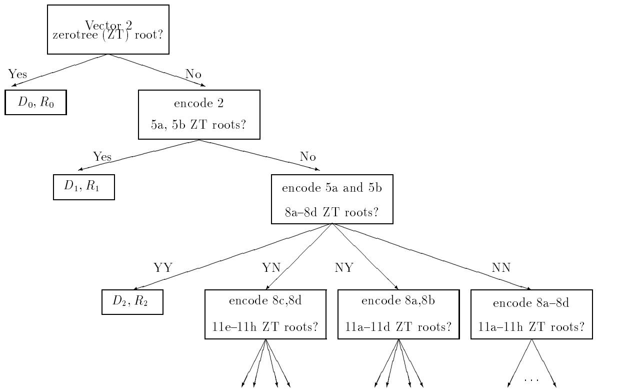

Figure 3: Tree of zerotree decisions (Y = Yes, N = No)

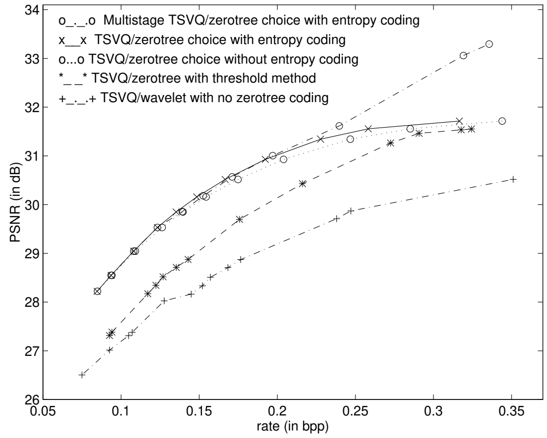

Figure 4: PSNR vs. bit rate for various wavelet/TSVQ algorithms

Figure 5: Original "Lena" image at 8 bpp

Figure 6: Encoded image at 0.148 bpp with a PSNR of 30.16 dB

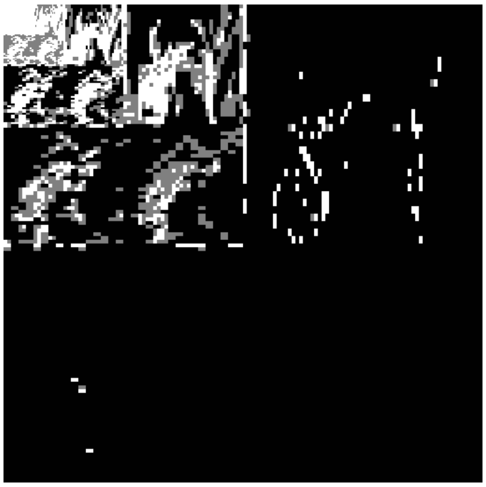

Figure 7: Significance map for Lena image at 0.148 bpp with zerotree choice algorithm

References

[1] P. C. Cosman, K. O. Perlmutter, and S. M. Perlmutter. Tree-structured vector quantization with significance map for wavelet image coding. In J. A. Storer and M. Cohn, editors, *Proceedings of the 1995 IEEE Data Compression Conference (DCC)*, pages 33–41, Snowbird, Utah, March 1995.

[2] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic Press, Boston, 1992.

[3] T. Senoo and B. Girod. Vector quantization for entropy coding of image subbands. *IEEE Transactions on Image Processing*, 1(4):526–532, October 1992.

[4] J. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Transactions on Signal Processing*, 41(12):3445–3462, December 1993.

[5] Z. Xiong, K. Ramchandran, and M. T. Orchard. Joint optimization of scalar and tree-structured quantization of wavelet image decomposition. In R. Madon, editor, *Proceedings of the Twenty-Seventh Annual Asilomar Conference on Signals, Systems and Computers*, pages 891–895, Pacific Grove, CA, November 1993.