GENERALIZED BIT-PLANES FOR EMBEDDED CODES

Eduardo A. B. da Silva

PEE/COPPE/DEL/EE
Universidade Federal do Rio de Janeiro
Cx. P. 68504, Rio de Janeiro, RJ
21945-970, BRAZIL
eduardo@lps.ufrj.br

Marcos Craizer

Departamento de Matemática
PUC-Rio
R. Marquês de São Vicente, 225
Rio de Janeiro,RJ
22453-900, BRAZIL
craizer@saci.mat.puc-rio.br

ABSTRACT

In many modern image compression techniques, the coefficients of an image transform are encoded by successive refinements. This is in general equivalent to successively encoding each bit-plane of the image transforms. Such methods, especially when the wavelet transform is employed, are among the state-of-the-art in image coding. In fact, several proposals to the JPEG 2000 standard use some sort of bit-plane encoding. Bit-plane encoding has been traditionally used as a scalar quantization technique, that is, each coefficient in individually decomposed in bit-planes. In this paper we analyze extensions of the bit-plane encoding concept to vectors, whereby each vector of coefficients is decomposed in "vector bit-planes". First, we formally describe the vector bit-plane representations, and then state a theorem concerning conditions for their existence. Next, we propose a modification to the proposed vector bit-plane representations and state a second theorem, which shows that the proposed modifications lead to much more robust algorithms. Simulation results are presented for the embedded encoding of wavelet coefficients of images, which confirm the potential advantages of vector over scalar bit-plane representations. These strongly indicate that vector bit-plane representations should be further investigated.

1. INTRODUCTION

Wavelet transforms have been widely investigated for image coding applications. Among the wavelet image coding methods, the ones that are based on bit-plane encoding have become very popular. In these methods, the wavelet coefficients are coded in successive passes. In each pass, one bit-plane of the wavelet coefficients is encoded. In general, the similarities of the bands of same orientation is taken into consideration in the encoding of the bit-planes, by the use of zero-trees or similar structures. Good examples of bit-plane wavelet coders can be found in [1, 2, 3]. Although not restricted to wavelet coders, [4] is also a nice example of this recent trend to the use of bit-plane encoding. The performances of wavelet coders based on bit-plane encoding places them among the state-of-the-art in image coding.

Besides the good performance that can be obtained with bit-plane encoding of wavelet coefficients, bit-plane encoders have the advantage of naturally generating an embedded bitstream. In other words, an embedded bitstream with, for example, 1 bit/pixel, contains all the bitstreams with less than 1 bit/pixel. If we partially decode it up to 0.5 bit/pixel, we would obtain the same image that would be obtained if a bitstream of 0.5 bit/pixel was generated and decoded in the first place.

Bit-plane encoding is a form of scalar quantization, because each coefficient is individually decomposed into bit-planes. However, coding efficiency mandates that each bit-plane be encoded considering sets of coefficients. Zero-trees [1], run-length coding [3] and arithmetic encoding using conditional probabilities [5] are examples of that. Therefore, it is natural to wonder if there are non-trivial and efficient generalizations of bit-plane encoding to vectors. In other words, are there efficient ways of combining the advantages of bit-plane encoding and vector quantization? The answer to this question is affirmative, and in fact [6] describes a coder consisting of a straightforward substitution of the "scalar" bit-plane encoder in [1] by a kind of "vector" bit-plane encoder. The coder in [6] has shown an encouraging improvement in performance over the one in [1].

In this paper, first the general problem of bit-plane encoding of vectors is analyzed. Then, two novel theorems which establish sufficient conditions for the vector bit-plane encoding be possible are stated, and some of its properties are analyzed. Based on these theorems, some improvements to the coder in [6], are proposed. Last, simulation results are presented, and extensions to this work are proposed.

2. VECTOR BIT-PLANES

To decompose a scalar quantity $-1 \leq c \leq 1$ using bit-planes is equivalent to represent it through a sequence $\{b_1, b_2, \ldots, b_n, \ldots\}$ such that:

$$c = s \sum_{i=1}^{\infty} b_i 2^{-i} \tag{1}$$

where $s \in \{-1, 1\}$ represents the sign of c and $b_i \in \{0, 1\}$.

Alternatively, we can have s always equal to 1 and $b_i \in \{1, -1\}$, yielding the representation below:

$$c = \sum_{i=1}^{\infty} b_i 2^{-i} \tag{2}$$

For example, in [2], $b_i \in \{0, 1\}$, and a representation like the one in eq. 1 is used. In [1], $b_i \in \{1, -1\}$, and coefficients are represented as in eq. 2.

In coding applications, the summation in eq. 2 is obviously not infinite, but goes from 1 to P, the number of bit planes, yielding the approximation c_P. In general, the more bit-planes are added to the summation, the smaller is the distortion $|c - c_P|$ in the representation of c. P is often chosen as the smallest value such that a certain distortion criterion is met, i.e. $|c - c_P| \leq \Delta$.

A trivial way to do vector bit-plane encoding of an N-dimensional vector $\mathbf{v}$ is to simply represent each of its coordinates v_k, for $k = 1, \ldots, N$ using bit-planes. Therefore, from eq. 2, we have that the vector $\mathbf{v}$ can be represented as:

$$\mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{\infty} b_{1i} 2^{-i} \\ \sum_{n=1}^{\infty} b_{2i} 2^{-i} \\ \vdots \\ \sum_{i=1}^{\infty} b_{Ni} 2^{-i} \end{pmatrix} \tag{3}$$

Eq. 3 is equivalent to:

$$\mathbf{v} = \sum_{i=0}^{\infty} \begin{pmatrix} b_{1i} \\ b_{2i} \\ \vdots \\ b_{Ni} \end{pmatrix} 2^{-i} = \sum_{i=0}^{\infty} \mathbf{b}_i 2^{-i} \tag{4}$$

We can see that eq. 4 is a vector version of eq. 2, that is, every vector $\mathbf{v}$ can be represented by a sequence of vectors $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n, \ldots\}$. If $b_{ki} \in \{1, -1\}$, $\mathbf{b}_i$ belongs to the codebook T_N whose vectors are of the form $\{((-1)^{\beta_1}\ (-1)^{\beta_2} \ldots (-1)^{\beta_N})^t\}$. Since all vectors b_i have magnitude equal to $\sqrt{N}$, they are located on an hyper-sphere, thus representing different orientations. For this reason, T_N can also be referred to as an *orientation codebook.*

Eq. 4 means that every vector $\mathbf{v}$ whose components are smaller than 1 can be represented as series of vectors of decreasing magnitudes (2^{-i}) and orientations drawn from a fixed orientation codebook T_N. As in the scalar case, the summations are not infinite, and go from 1 to the number of bit-planes P, yielding an approximation $\mathbf{v}_P$ of $\mathbf{v}$ as below:

$$\mathbf{v_P} = \sum_{i=0}^{Q} \mathbf{b}_i 2^{-i} \tag{5}$$

In coding applications, what is wanted is the smallest possible P (or, more exactly, the smallest possible entropy of the ensemble of vectors $\mathbf{b}_i$) such that the distortion $\|\mathbf{v} - \mathbf{v}_P\| \leq \Delta$.

In this trivial extension of bit-plane encoding to vectors, the orientation codebook is composed by vectors whose components are 1 and -1 (eq. 4). At this point, a natural question to ask is whether there are other orientation codebooks such that vector bit-plane encoding is more efficient. More precisely, we are looking for representations of a vector $\mathbf{v}$ having the form:

$$\mathbf{v}_Q = \sum_{i=1}^{Q} \mathbf{u}_{n_i} \alpha^i \tag{6}$$

where $\mathbf{u}_{n_i} \in C_N$, an orientation codebook composed by unitary vectors on an hyper-sphere. We should also observe that the terms 2^{-i} in eq. 4 have been replaced by terms α^i, $0 < \alpha < 1$.

We want $\mathbf{v}_Q$ to arbitrarily approximate $\mathbf{v}$ for Q sufficiently large, that is,

$$\lim_{Q \to \infty} \mathbf{v}_Q = \mathbf{v} \tag{7}$$

Supposing that P and Q are the minimum number of terms in eqs. 5 and 6, respectively, such that the respective approximation errors are smaller than Δ. We have that the representation in eq. 6 is more efficient than the one in eq. 4 if the entropy of the ensemble of vectors $\mathbf{b}_i$, $i = 1, \ldots, P$ is larger than the one of the ensemble of vectors $\mathbf{u}_{n_i}$, $i = 1, \ldots, Q$, and vice versa.

In order to investigate more efficient vector bit-plane representations, one must first determine under what conditions does a general vector bit-plane representation as in eq. 6 exists such that eq. 7 holds. This will be dealt with in the next section.

3. EXISTENCE OF VECTOR BIT-PLANE REPRESENTATIONS

In order to determine conditions for the decompositions in eq. 6 to satisfy eq. 7 we have to first define one important parameter from an N-dimensional orientation codebook C_N, which we refer to as $\Theta(C_N)$[1]. It is the maximum possible angle between any vector $\in \mathbb{R}^N$ and its nearest neighbor $\in C_N$. More precisely,

$$\Theta(C_N) = \cos^{-1}\{\min_{\mathbf{x}\in\mathbb{R}^N}\{\max_{\mathbf{u}_i\in C_N}\{\frac{\mathbf{x}\cdot\mathbf{u}_i}{\|\mathbf{x}\|\|\mathbf{u}_i\|}\}\}\} \quad (8)$$

We have then the following theorem:

Theorem 1 *Given an orientation codebook $C_N = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_M\}$ such that $\|\mathbf{u}_i\| = 1$, $\forall i$, then there exists a representation as in eq. 6 such that eq. 7 is valid for all $\mathbf{v} \in \mathbb{R}^N$, $\|\mathbf{v}\| \leq 1$ if:*

$$\frac{1}{2\cos[\Theta(C_N)]} \leq \alpha < 1, \quad \Theta(C_N) \leq 45^\circ \quad (9)$$

$$\sin[\Theta(C_N)] \leq \alpha < 1, \quad \Theta(C_N) \geq 45^\circ \quad (10)$$

It is important to point out that this theorem only establishes sufficient conditions for eq. 7 to be valid for a representation as in eq. 6. In fact, its proof supposes worst case conditions. By worst case conditions it is meant that the angle between the residual $\mathbf{e}_i = \mathbf{v} - \mathbf{v}_i$ and $\mathbf{u}_{n_{i+1}}$ is supposed always to be equal to $\Theta(C_N)$, which is clearly very pessimistic.

An important property that can be deduced from eqs. 9 and 10 is that the smaller the value of $\Theta(C_N)$, the smaller α can be. We can estimate the impact of the values of α in eq. 6 by noting that, if eq. 7 is valid, the magnitude of the error of a Q-term approximation, $\mathbf{e}_Q = \mathbf{v} - \mathbf{v}_Q$ is given by

$$\|\mathbf{e}_Q\| = \|\sum_{i=Q+1}^{\infty} \mathbf{u}_{n_i}\alpha^i\| \leq |\sum_{i=Q+1}^{\infty} \alpha^i| = \frac{\alpha^{Q+1}}{1-\alpha} \quad (11)$$

Eq. 11 shows us that the smaller the value of α, the smaller will be the bound on the residual approximation error. Therefore, from a coding point of view, it is interesting to have the smallest possible value of α in order to minimize the approximation error. Then, from eqs. 9 and 10 we have that the orientation codebook C_N should be such that value of $\Theta(C_N)$ is as small as possible. For any given dimension, there are two ways of reducing the values of $\Theta(C_N)$: (i) by increasing the number M of vectors; (ii) by distributing the vectors "more uniformly" over the N-dimensional unity sphere. However, when the number M of vectors in C_N is increased, there is a compromise: despite the decrease in the value of $\Theta(C_N)$, and, consequently, of α and the truncation error using Q vectors, there will be an increase in the entropy of the set $\mathbf{u}_{n_i}$, $i = 1, \ldots, Q$. Therefore, for the entropy to be maintained Q would have to be reduced, thereby increasing the distortion. Thus, the best way to have a codebook with a low value of $\Theta(C_N)$ is to have its vectors the more "uniformly distributed" possible over the unity sphere in $\mathbb{R}^N$ for a given number M of vectors. Good examples of codebooks satisfying this property are the first shells of the lattices which solve the sphere packing problem in N dimensions [7]. For example, analyzing eq. 4, which describes the case of a mere concatenation of the bit-planes of the vector components, it can be proven that the codebook T_N has $\Theta(T_N) = \cos^{-1}(\sqrt{1/N})$. Table 1 compares the number of vectors and the values of $C(T_N)$ with the ones from the first shells of the lattices D_4, E_8 and Λ_{16}, which solve the sphere packing problem in dimensions 4, 8 and 16.

Codebook	T_4	D_4	T_8	E_8	T_{16}	Λ_{16}
M	16	24	256	240	65536	4320
Θ	60°	45°	69°	45°	76°	55°

Table 1: Values of Θ and number M of vectors for several orientation codebooks.

From this table we can see clearly the superiority of the codebooks derived from the lattices that solve the sphere packing problem. For example, T_8, despite having more vectors than E_8, has a much larger value of Θ. This implies that its vectors are much less uniformly distributed than the ones of E_8, and therefore vector bit-plane representations based on it are less efficient.

An algorithm for computing vector bit-plane representations

Theorem 1 described conditions for the existence of vector bit-plane representations as in eq. 6 satisfying eq. 7. An important point when it comes to practical applications is whether there is a fast algorithm for computing such representations. Fortunately, the answer is affirmative, and is given by the following greedy algorithm[2]:

[1] In [6] it has been referred to as $\theta_{\max}$.

[2] It should be noted that is is being assumed, without loss of generality, that $\|\mathbf{v}\| \leq 1$.

1. Make $m = 0$, $\mathbf{e}_0 = \mathbf{v}$ and $\beta = \alpha$.

2. Given the vector $\mathbf{e}_m$, choose $i_{m+1} \in \{1, \dots, M\}$, where M is the size of the orientation codebook C_N, such that:

$$\mathbf{e}_m \cdot \mathbf{u}_{i_{m+1}} = \max\{\mathbf{e}_m \cdot \mathbf{u}_k : 1 \leq k \leq M\}$$

3. Compute $\mathbf{e}_{m+1} = \mathbf{e}_m - \beta \mathbf{u}_{i_{m+1}}$

4. Increment m, multiply β by α and go to step 2

One should note that this algorithm determines, in each pass, the vector $\mathbf{u}_{i_{m+1}}$ which is closest to the residual $\mathbf{e}_m$, and therefore minimizes the error in the representation of $\mathbf{e}_m$ in that pass. However, this procedure is not guaranteed to generate the optimum representation, that is, the one which yields the minimum representation error after Q passes. More precisely, if the algorithm above generates a sequence of vectors $u_{i_1}, u_{i_2}, \dots, u_{iQ}$, which, according to eq. 6, provides an approximation $\mathbf{v}_{1Q}$, there is no guarantee that there is not a different sequence of vectors $u_{j_1}, u_{j_2}, \dots, u_{jQ}$ providing an approximation $\mathbf{v}_{2Q}$ according to eq. 6 such that $||\mathbf{v} - \mathbf{v}_{2Q}|| < ||\mathbf{v} - \mathbf{v}_{1Q}||$. In other words, this discussion implies that, besides the fact that a representation as in eq. 6 satisfying eq. 7 is not unique, we have that the above algorithm will not necessarily find the optimum one. Fortunately, in most cases, the approximation it finds performs well enough.

At this point it is instructive to point out that the above algorithm has some similarities to Mallat's matching pursuit algorithm [8]. The main difference is that in Mallat's matching pursuits we replace the α^i term in eq. 6 by the projection of $\mathbf{e}_{i-1} = \mathbf{v} - \mathbf{v}_{i-1}$ on $\mathbf{u}_{n_i}$. More precisely, after Q passes, a matching pursuit decomposition of a vector $\mathbf{v}$ is of the form:

$$\mathbf{v}_Q = \sum_{i=1}^{Q} \gamma_i \mathbf{u}_{n_i} \quad (12)$$

where, likewise eq. 6, $\mathbf{u}_{n_i} \in C_N$, an orientation codebook composed by unitary vectors on an hyper-sphere. On the other hand, unlike eq. 6,

$$\gamma_i = (\mathbf{v} - \mathbf{v}_{i-1}) \cdot \mathbf{u}_{n_i} \quad (13)$$

This implies that, while in the vector bit-planes representation, a vector is represented by just a sequence of unity vectors $u_{i_1}, u_{i_2}, \dots$, in Mallat's matching pursuits a vector is represented by a sequence of unity vectors $u_{j_1}, u_{j_2}, \dots$ plus a sequence of projections $\gamma_{j_1}, \gamma_{j_2}, \dots$

Performance of vector bit-plane encoding in the context of embedded wavelet coding

The vector bit-plane decomposition described above has been used in place of the conventional bit-plane decomposition in an EZW-like [1] algorithm. Details can be found in [6]. The results there have shown a performance improvement of around 1 dB for the LENA 512×512 image in the vicinity of 0.5 bit/pixel when the lattice Λ_{16} is used. As expected from the above discussions, the performance of the algorithm varies a great deal with α. For example, with Λ_{16}, the PSNR performance reaches a reasonably pronounced peak for α around 0.62 (see the "conventional algorithm" results in figure 1. This is in accord with what has been discussed above in relation with Theorem 1, because if α is too small eq. 7 is not valid, that is, the magnitude of the error $\mathbf{e}_Q$ does not tend to zero as $Q \to \infty$. If α gets too large, despite eq. 7 being valid, $||\mathbf{e}_Q||$ increases (see eq. 11). The main problem with this sort of behaviour is that the α value which gives peak performance is image dependent. That is inconvenient in a number of applications, especially when encoding time should be kept as low as possible.

In the next section we propose another theorem, together with a modification to the vector bit-plane decomposition which solves this problem, and provides almost peak performance for a large range of α values.

4. A MODIFIED VECTOR BIT-PLANE ENCODER

It can be shown that, in order for eq. 7 to hold for any $0 < \alpha < 1$, the algorithm above has to be modified in order to guarantee that the residual in pass m is such that[2] $\alpha^{m+1} \leq ||\mathbf{e}_m|| \leq \alpha^m$. This is done as follows: First, the zero vector is added to the codebook C_N. If the magnitude of $\mathbf{e}_m$ is smaller than α^{m+1}, $\mathbf{u}_{i_{m+1}}$ is chosen to be the zero vector, so that there is no refinement for that pass. If the magnitude of $\mathbf{e}_m$ is greater than α^m, β (see step 3) will not be multiplied by α in that pass. In a practical algorithm this can be signalized to the decoder by the inclusion of an escape code before the vector for that pass. With this modification, eq. 6 becomes:

$$\mathbf{v} = \sum_{i=1}^{\infty} \left(\sum_{j=1}^{p(\mathbf{v},i)} \mathbf{u}_{n_{i_j}} \right) \alpha^i \quad (14)$$

Then, a new theorem can be stated, stronger then Theorem 1:

Theorem 2 *Suppose that the orientation codebook used in vector bit-plane encoding has $\Theta(C_N) \leq 60°$. Then* *a decomposition such as the one in eq. 14 exists for every* $0 < \alpha < 1$.

It is important to notice that, in order for the vector bit-plane decomposition in eq. 14 to be practical, $p(\mathbf{v}, i)$ should be small with great probability. For this reason, we used only $0.5 < \alpha < 1$ in our experiments. Indeed, we observed in these experiments that $p(\mathbf{v}, i) = 1$ occurred with a probability near 1 for these values of α.

Another point is that, likewise the algorithm derived from Theorem 1, this new algorithm does not necessarily lead to an optimal representation, but its performance is sufficiently good.

Experimental results

We have implemented the modifications leading to Theorem 2 to the vector bit-plane coder in [6]. The performance of this improved algorithm and the algorithm in [6] are compared in figure 1 for $0.5 \leq \alpha < 1$ using the first shell of the lattice Λ_{16} as the orientation codebook, for three different images. It can be observed that in the improved algorithm the performance does not degrade when α decreases, and therefore the choice of α is much less critical than in the previous case. In practice, this means that the value of α and can be made image independent, and therefore the modified algorithm is much more robust.

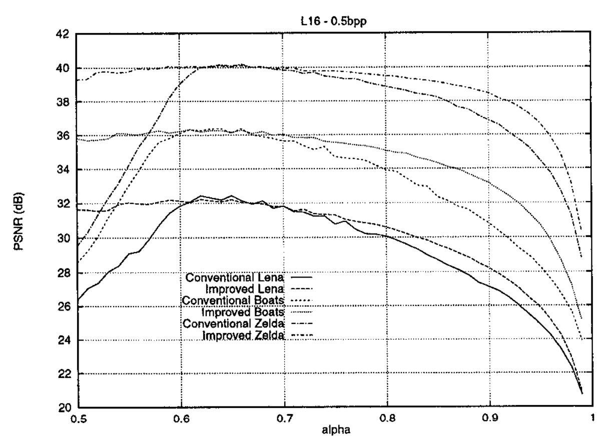

Figure 1: $\alpha \times PSNR$ for the images ZELDA, BOATS and LENA 256×256 at 0.5 bit/pixel with the "conventional" [6] and "improved"' (theorem 2) vector bit-plane coder, using the first shell of Λ_{16} as orientation codebook.

5. CONCLUSIONS

In this paper the concept of bit-plane encoding has been extended to vectors. It has been shown that there are orientation codebooks which can provide better performance than the trivial codebooks formed by separately bit-plane encoding each component of the vectors. Two theorems concerning the existence of "good" vector bit-plane decompositions have been stated. It has also been shown that when the first shells of the regular lattices which solve the sphere packing problem are used as orientation codebooks, the variations of vector bit-plane encoding proposed in this paper can outperform the conventional bit-plane encoding in EZW-like embedded wavelet encoders. However, the optimum choice of orientation codebooks still needs further investigation. Considering the great importance of bit-plane encoding in many modern image coding schemes, the results obtained are encouraging.

6. REFERENCES

[1] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 41, pp. 3445–3462, December 1993.

[2] A. Said and W. A. Pearlman, "A new, fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, pp. 243–250, June 1996.

[3] J. M. Tsai, J. Villasenor, and F. Chen, "Stack-run image coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, pp. 519–521, October 1996.

[4] J. Andrew, "A simple and efficient hierarchical image coder," in *1997 IEEE International Conference on Image Processing*, (Santa Barbara, CA), October 1997.

[5] R. V. Algazi and R. E. Jr., "Analysis-based coding of image transforms and subband coefficients," in *Proceeding of the SPIE*, pp. 11–21, 1995.

[6] E. A. B. da Silva, D. G. Sampson, and M. Ghanbari, "A successive approximation vector quantizer for wavelet transform image coding," *IEEE Transactions on Image Processing, Special Issue on Vector Quantization*, vol. 5, pp. 299–310, February 1996.

[7] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*. New York: Springer-Verlag, 1988.

[8] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, pp. 3397–3415, December 1993.