IMAGE CODING USING SUCCESSIVE APPROXIMATION WAVELET VECTOR QUANTIZATION

*E. A. B. da Silva** *D. G. Sampson* *M. Ghanbari*

Department of Electronic Systems Engineering – University of Essex
Wivenhoe Park – Colchester CO4 3SQ – England
Tel: +44 206 872448 Fax: +44 206 872900
e-mails: dasie@essex.ac.uk, sampd@essex.ac.uk, ghan@essex.ac.uk

ABSTRACT

A novel coding method of wavelet coefficients of images using vector quantization, referred as Successive Approximation Wavelet Vector Quantization (SA-W-VQ) is proposed. In this method, each vector is coded by a series of vectors of decreasing magnitudes until a certain distortion level is reached. Analysis of the successive approximation using vectors is given, and conditions for convergence are derived. It is shown that lattice codebooks offer an efficient tool to meet these conditions, with the extra advantage of fast encoding algorithms. In SA-W-VQ, distortion equalization of the wavelet coefficients can be achieved together with high compression ratio and precise bit rate control. Simulation results for still image coding show that SA-W-VQ outperforms both the EZW coder [1] and the standard JPEG.

1. INTRODUCTION

Wavelet transforms have been attracting the image coding community in the last few years. They are mainly used to decorrelate the image data, so that the resulting coefficients can be efficiently coded.

Besides the decorrelation of the image data, two dimensional wavelet transforms have another important property. Despite the low correlation among themselves, bands of same orientation look like scaled versions of each other. That is, their edges are approximately in the same corresponding positions, and hence, their non-significant coefficients are approximately in the same corresponding locations. In the coding of the wavelet coefficients, this similarity can be exploited to provide an efficient addressing of the non-zero coefficients, by generating zero-tree roots [1]. In fact, any efficient wavelet coding technique should exploit this property.

Since the wavelet transform is equivalent to an octave band subband decomposition, one of its advantages is that each band of coefficients corresponds to a certain frequency band. Thus, the bit rate can be allocated to the bands such that coefficients corresponding to frequency bands to which the eye is more sensitive have less quantization distortions (*noise shaping*). This, however, imposes some restrictions on the encoders, for they have to set up specific relative levels of distortion among the bands in order for them to match with the human visual system (HVS) sensitivity for each band.

Another property of wavelet transforms is that a quantization error introduced to a coefficient will appear as a scaled version of the synthesis wavelet superimposed on the reconstructed signal. In image coding, this implies that a quantization error from a coefficient located, for example, at an edge, will not be just confined to that edge, but will be spread through the reconstructed image with the shape of the corresponding wavelet, causing an annoying ringing. Therefore, the edge masking effects cannot easily be exploited when coding the coefficients, and the quantization distortion in every individual coefficient can be important to the final image quality.

One way to achieve the last two requirements is to code the wavelet coefficients in successive passes, whereby in each pass the quantization error is further refined, and a maximum error in each coefficient can be guaranteed. Since this also approximately guarantees an average level of distortion for each band, noise shaping can be readily obtained by properly weighting the bands prior to coding.

Successive approximation of the wavelet coefficients with scalar quantization, together with zero-tree coding, has been used in the embedded zero-tree wavelet (EZW) coder [1]. In this paper we introduce a successive approximation vector quantization scheme for coding of wavelet coefficients. It satisfies all the requirements above while exploiting the advantages of vector quantization. Simulation results show that the successive approximation wavelet vector quantisation achieves better performance than EZW for still image coding.

2. VECTOR SUCCESSIVE APPROXIMATION

2.1. Definition of the problem

Successive approximation in the scalar case is equivalent to approximation of a given length L by using at each pass yardsticks of increasingly smaller lengths, until a certain level of error is obtained. Generalization of this process to k-dimensional space is not a straightforward task. A k-dimensional vector can be defined by two parameters : its length, which is a scalar value that corresponds to the norm

*His work was supported in part by Universidade Federal do Rio de Janeiro and Conselho Nacional de Desenvolvimento Científico e Tecnológico, Brazil, under grant 200885/91-0

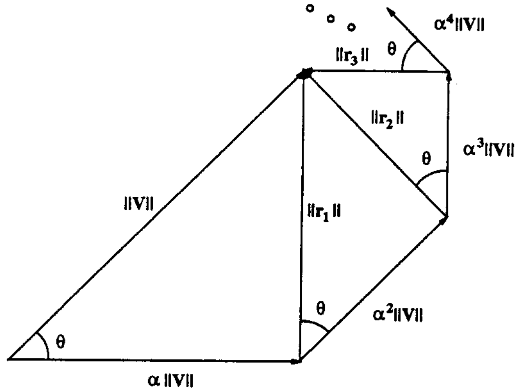


Figure 1: Analysis of convergence for the n-dimensional case

of the vector and its orientation, which is a k-dimensional vector with unit energy. Therefore, in vector successive approximation, unlike the scalar case, instead of yardsticks of decreasing lengths, we deal with "vector yardsticks" having decreasing lengths and given orientations in a k-dimensional space. In practice, these "vector yardsticks" will be chosen from a finite codebook, and therefore the set of possible orientations is finite. At each pass, the "vector yardsticks" will aim to approximate the residual vector formed as the difference between the original vector $\vec{V}$ and its approximation so far. The question which naturally arises is: how can one guarantee that the vector approximation process converges, that is, if the number of passes is sufficiently large, the magnitude of the residual vector will be always smaller than an arbitrarily value.

2.2. Conditions for convergence

In this section we devise sufficient conditions for the convergence of the successive approximation of vectors using a finite set of orientation vectors of decreasing length. The following suppositions have been made :

(1) The *orientation codebook*, $\mathbf{Y}$, is a finite set of k-dimensional vectors with unit energy. At each pass, a new "vector yardstick" is formed as the product of the current yardstick length and one of the unit orientation code vectors $\vec{y}_i$.

(2) The orientation codebook is built so that the solid angle between any possible vector and its closest orientation codevector is upper bounded by $\theta_{\max}$. Thus, at each pass the maximum error is introduced when the residual vector is approximated by a vector with orientation $\theta_{\max}$.

(3) The yardstick length at each pass will be scaled by a constant factor α, so-called *approximation scaling factor*, in the range of $0.5 \leq \alpha < 1.0$.

(4) For a given vector $\vec{V}$, the approximation process is activated for the first time at a certain pass indexed by i, if and only if $\ell_i < \|\vec{V}\| \leq \ell_i/\alpha$. ℓ_i denotes the yardstick length at pass i and $\|\vec{V}\|$ is the norm of the vector $\vec{V}$. Therefore, the maximum error introduced by the first pass will occur in the case that $\|\vec{V}\| = \ell_{i-1} = \ell_i/\alpha$.

The sufficient conditions to guarantee the convergence of the successive approximation by a finite set of orientation vectors of decreasing lengths can be derived by evaluating the worst case. Figure 1 illustrates this process. From the supposition **(2)**, for each pass i, the maximum error in the orientation is equal to $\theta_{\max}$ and from the supposition **(4)**, the maximum error in the length will occur if the initial yardstick length is set equal to $\ell_i = \alpha\|\vec{V}\|$. From figure 1, defining $\|\vec{r}_0\| = \|V\|$, the magnitude of the residual vector after n passes is given by:

$$\|\vec{r}_n\|^2 = \|\vec{r}_{n-1}\|^2 + \alpha^{2n}\|\vec{V}\|^2 - 2\|\vec{r}_{n-1}\|\alpha^n\|\vec{V}\|\cos(\theta_{\max})$$

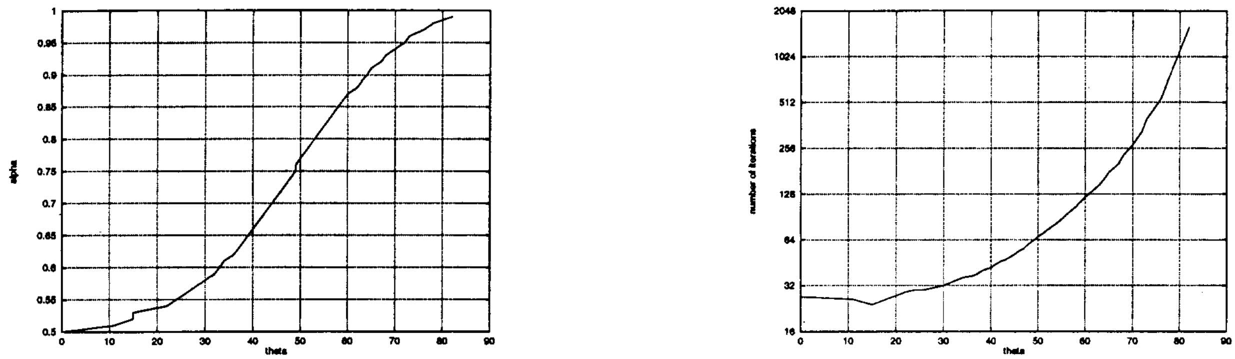
With the above recursive formula we can compute the residual vector magnitudes after each pass ($\|\vec{r}_i\|$) for any given pair $(\alpha, \theta_{\max})$. We have used it to find the value of the *convergence scaling factor* $\bar{\alpha}$, for any $\theta_{\max}$ in the range $0° \leq \theta_{\max} < 90°$, such that the scheme converges* for any $\alpha \geq \bar{\alpha}$, where $0.5 \leq \alpha < 1.0$. Figure 2.a gives the values of the convergence scaling factor $\bar{\alpha}$ for all angles $\theta_{\max}$ in the range $0° \leq \theta_{\max} < 90°$. Figure 2.b shows $\theta_{\max}$ against the number of iterations required for convergence when $\alpha = \bar{\alpha}$. The results illustrated in figure 2 show that, for $\theta_{\max}$ up to around 80° this successive approximation scheme is guaranteed to converge, provided that a suitable value of α is chosen. For $\theta_{\max} = 0°$, which is equivalent to the scalar case, convergence is guaranteed for $\alpha = 0.5$. For this value, the number of iterations is minimum. As $\theta_{\max}$ increases, so do both α and the number of iterations. From a compression point of view, more iterations would require more bits to achieve a certain distortion. Hence, the selected orientation codebook should be such that $\theta_{\max}$ is as small as possible. Nevertheless, there is a compromise between the value of $\theta_{\max}$ and the resolution of the orientation codebook, determined by $\frac{\log_2 N}{k}$, where N is the codebook population and k is the vector dimension. In one extreme, we have $\theta_{\max} = 0°$ for the scalar case. However, if vectors of higher dimension are used, there can be gains in bit rate despite the larger values of $\theta_{\max}$ and number of iterations, due to the savings of vector over scalar quantizers.

2.3. Lattice based orientation codebooks for successive vector approximation

After determining the necessary conditions for converge of the successive approximation scheme, we discuss the requirements of a "good" orientation codebook selection. We need to guarantee that the maximum error in orientation introduced by approximation at any stage is bounded by a well defined value $\theta_{\max}$. Moreover, the graph in figure 2.b indicates that $\theta_{\max}$ should be as small as possible, such that the fastest possible convergence to an arbitrary error is achieved.

Therefore, the main requirement in the design of the orientation codebook is to provide a fairly low value of $\theta_{\max}$, regardless of the individual vector locations in the

*convergence is assumed when the improvement in the approximation after two subsequent passes is less than 10^{-8} of the magnitude of the original vector.



(a) (b)

Figure 2: Plots of θ versus: (a) $\bar{\alpha}$; (b) number of iterations required for convergence.

k-dimensional space. Lattice codebooks are a good choice for the orientation codebook of the proposed scheme, because they can offer a good trade-off between $\theta_{\max}$ and the codebook population due to their space packing properties. Lattice codebooks also offer the advantage of simple and fast encoding algorithms [2] .

In general, the points of a given regular lattice are distributed on the surface of successive, concentric, k-dimensional *hyper-shells* centered at the origin, so that all lattice points at the same shell have the same l_r-norm. In our experiments, the m^{th} spherical ($r = 2$) shell $S_m(L_k)$ of a given lattice L_k is the set of all L_k-points at the same euclidian distance from the origin.

Assuming that the orientation codebook is built by the lattice points from shell S_m of a given lattice L_k, the following parameters play a key role in the evaluation of the efficiency of a particular lattice orientation codebook:

(i) the solid angle between the nearest neighbor code vectors $\theta_{NN}(L_k, S_m)$;

(ii) the population of the lattice points on the particular shell $N_m(L_k, S_m)$;

(iii) the maximum possible angle between any input vector and its closest codevector $\theta_{\max}(L_k, S_m)$;

Lattices are associated with the best known sphere packings in k-dimensional space [3]. In general, the nearest neighbor angle $\theta_{NN}(L_k, S_m)$ for a given lattice L_k and a spherical shell S_m, can be calculated only from the radius of the lattice packing $\rho(L_k)$ and the energy of the particular shell $r(L_k, S_m)$ [4], and the maximum possible angle $\theta_{\max}$ between any input vector and its closest codevector has been computed exhaustively. Table 1 summarizes the parameters of regular lattices which give the best lattice packing at dimensions k = 4, 8 and 16. The orientation codebooks used and tested with our coding scheme are built based on these lattices.

lattice type, L_k	shell index, m	popul., N_m	nearest neighbor θ_{NN}	maximum angle, $\theta_{\max}$
D_4	1	24	60°	44°
D_4	2	24	60°	44°
D_4	1+2	48	45°	31°
E_8	1	240	60°	42°
E_8	2	2160	41°	41°
E_8	3	6720	33°	31°
E_8	1+2	2400	41°/33°	30°
E_8	1+2+3	9120	30°/35°	25°
Λ_{16}	2	4320	60°	47°

Table 1: Parameters of the regular lattices with best packing in dimensions $k = 4, 8, 16$

3. A SUCCESSIVE APPROXIMATION VECTOR WAVELET CODER

In this section we introduce a new method for wavelet image coding based on successive approximation vector quantization, described in section 2.1. We refer to this method as *Successive Approximation Wavelet Vector Quantization* (SA-W-VQ).

In coding of images with SA-W-VQ, first the image mean is extracted. An R stage biorthogonal wavelet transform is then applied to the zero-mean image. Each band is normalized so that the distortion equalization provided by the successive approximation process is equivalent to a noise shaping according to certain HVS response [5]. In the simulation results presented, though, the HVS response was assumed to be flat.

Each band is then divided into M-dimensional vectors. The maximum magnitude V of all the vectors is then computed. Initially, the yardstick length ℓ is set to αV, where the value of α is chosen according to the $\theta_{\max}$ value of the selected lattice codebook. All the vectors are scanned, and the ones with magnitude smaller than ℓ are set to zero. Each of the remaining vectors is replaced by its closest orientation codevector scaled with a magnitude ℓ. After this pass, the locations of the zero vectors are transmitted. This is done via 3 symbols: zero (Z), zero tree root (ZT) and coded value (C). If a vector is zero and all of its corresponding vectors in the higher bands of the same orientation are also zero this vector is replaced by a ZT, so that it is not necessary to transmit its corresponding vectors. Only for the lowest frequency band, a ZT implies that the corresponding vectors in all bands are zero. In case that a vector is zero but not a ZT, it is marked as Z, and no information can be inferred about its corresponding vectors. On the other hand, a non-zero vector is replaced by a coded value symbol (C).

The string generated by the three symbols (ZT, Z and C), is then encoded by the arithmetic coder described in [6] using an adaptive model. In the higher frequency bands, since there are no ZT's, the arithmetic coder uses a model with only 2 symbols (Z and C). After encoding this string, which indicates the locations of the zero vectors, the orientation code vectors of the non-zero vectors (marked as C) are encoded. For this purpose, the model of the arithmetic coder is reinitialized to have as many symbols as the population of the orientation codebook. The yardstick length ℓ is then updated through multiplication by α. The difference between the original and the non-zero reconstructed vectors is coded using the new yardstick length. The indexes of the new orientation code vectors are also encoded into the bitstream via the arithmetic coder. This whole process is repeated again until a certain bit rate is achieved.

The initial yardstick length, the value of α, the image means and some coding parameters are transmitted in the header. In our implementation the header is 10 bytes long.

4. EXPERIMENTS - SIMULATION RESULTS

In this section, the performance of SA-W-VQ with various lattice codebooks is compared against the standard JPEG and the EZW coder proposed by Shapiro [1]. The wavelet transform used throughout this work was a 5 stage octave band decomposition implemented by the filter bank 6.b5_ra7 described in [7]. We employed only monochrome version of the images. However, extension to color is straightforward.

In the first experiment, we evaluated the performance of the orientation codebooks based on the lattices given in Table 1. First, the best value of the approximation scaling factor α is estimated for each lattice codebook. They are, for example, 0.55 for D_4, 0.60 for E_8 and 0.62 for Λ_{16}. With the optimum values of α, Λ_{16} tends to give better performance compared to both E_8 and D_4 [4]. This is exemplified in Table 2 for a bit rate of 0.4 bit/pixel.

Table 2 summarizes the PSNR results obtained by these methods. These results demonstrate that the proposed coding scheme achieves considerably better R-D performance compared with the JPEG coder. SA-W-VQ also performs consistently better than the EZW coder, which is a very efficient successive approximation wavelet coder for the scalar case.

5. CONCLUSIONS

A method of performing successive approximation of wavelet coefficients using vector quantization (SA-W-VQ) was introduced. An analysis of the convergence of the successive approximation vector quantization (SA-VQ) was made. It was found that the most important feature of a SA-VQ codebook is the maximum possible error in orientation when an input vector is represented by its closest codevector, which makes orientation codebooks based on regular lattices particularly suitable for SA-VQ.

Test Image	D_4	E_8	Λ_{16}	EZW	JPEG
BARBARA	29.36	30.60	30.90	29.03	27.27
BOATS	34.19	34.78	35.24	34.29	32.63
GIRL	35.27	35.91	36.12	35.14	33.98
GOLD	31.01	32.76	32.61	32.48	31.38
ZELDA	38.43	39.36	39.44	39.08	37.16
LENA 256	30.13	30.15	30.29	30.06	28.07
LENA 512	35.17	35.86	36.09	35.02	33.42

Table 2: PSNR performance of SA-W-VQ (dB) for several test images at a rate of 0.4 bit/pixel compared with the EZW [39] algorithm and JPEG

Several lattice codebooks were compared, and the first shells of the lattices were shown to give the best performance. Among these, the Barnes-Wall lattice Λ_{16} offered the best rate-distortion results.

SA-W-VQ has achieved very good rate-distortion results, comparable to those of the most successful methods reported in the literature. In addition, noise shaping can be implemented by simple weighting of the coefficients prior to coding. Moreover, SA-W-VQ has the advantage of a very simple encoding process due to the fast NN algorithms of the lattice codebooks.

6. REFERENCES

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

[2] D. G. Sampson and M. Ghanbari, "Fast lattice-based gain-shape vector quantisation for image sequence coding," *IEE Proceedings Part-I*, vol. 140, pp. 55–65, February 1993.

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

[4] E. A. B. da Silva, D. G. Sampson, and M. Ghanbari, "A successive approximation vector quantizer for wavelet transform image coding." Submitted to *IEEE Transactions on Image Processing*.

[5] B. Macq, "Weighted optimum bit allocations to orthogonal transforms for picture coding," *IEEE Journal on Selected Areas on Communications*, vol. 10, pp. 875–883, June 1992.

[6] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Communications of the ACM*, vol. 30, pp. 520–540, June 1987.

[7] E. A. B. da Silva and M. Ghanbari, "On the coding gain of wavelet transforms," in *1994 IEEE International Symposium on Circuits and Systems*, (London), pp. 3.193–3.196, June 1994.