SUCCESSIVE APPROXIMATION VECTOR QUANTIZATION WITH IMPROVED CONVERGENCE

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 - Successive approximation vector quantization (SA-VQ) is a relatively recent algorithm in which each vector is represented by a series of vectors of decreasing magnitudes and orientations drawn from a fixed orientation codebook. It has been shown to provide good performance in wavelet coding schemes. In this paper, analytical results concerning the convergence of SA-VQ are presented in the form of two theorems. In the first one, results which had been previously determined only experimentally are presented analytically. In the second, a modification is proposed to the original SA-VQ algorithm which improves its convergence properties. Then, image compression results deriving from the application of the modified SA-VQ algorithm to coding wavelet transform coefficients are presented, showing improved PSNR performance.

1. Introduction

Wavelet transforms have been increasingly used in image coding applications. Their multi-resolution properties, high coding gain and possibility of avoiding the annoying blocking artifacts commonly encountered in DCT-based methods make them a very attractive alternative for the transformation step in image compression systems. There is a plethora of methods of quantizing and coding wavelet transform coefficients. One class of such methods is composed by the ones that use successive approximation of the coefficients, that is, the coefficients are coded in successive passes, and in each pass the precision in their representation is increased. More specifically, after pass P, a scalar coefficient c would be approximated by c_P such that:

$$c_P = \sum_{n=1}^{P} s_n \tag{1}$$

Since the precision in the representation of c by c_P in equation 1 must increase with the number of passes P, the magnitudes of the scalars s_n must almost always decrease as n increases. In general, $s_n = B\beta_n\alpha^n$, where $0 < \alpha < 1$, $\beta_n \in \{-1, 0, 1\}$ and B depends on the dynamic range of the set of scalars being coded. A good example of successive approximation of scalars is the binary representation of real numbers, when $\alpha = 1/2$. When such type of successive approximation is used in image coding it is also referred to as *bit-plane encoding*.

When employed for coding wavelet coefficients, the successive approximation quantization methods provide performances among the best in the literature. In general, they are used in conjunction with the exploitation of the similarities among bands of same orientation, in the form of zerotrees. Good examples can be found in [1] and [2].

Considering the efficiency of successive approximation *scalar* quantization in coding wavelet coefficients, it is natural to wonder whether such methods could be extended to vectors in order to incorporate some of the advantages of vector over scalar quantizers. Residual vector quantization addresses this issue [3]. In it, a k-dimensional vector $\mathbf{v}$ can be, after P stages, approximated by $\mathbf{v}_P$ as follows:

$$\mathbf{v}_P = \sum_{n=1}^{P} \mathbf{w}_n \tag{2}$$

$\mathbf{w}_n$ is a k-dimensional vector drawn from the codebook of pass n, $\mathcal{Y}_n$. In general, the magnitudes of

0-7803-5030-8/98/$10.00 ©1998 IEEE

the vectors $\mathbf{w}_n$ decrease with n. An important disadvantage of this approach is that it is extremely expensive computationally to find optimum codebooks $\mathcal{Y}_n$ given a training set and number of stages P. Indeed, for coding wavelet coefficients, successive approximation scalar quantization performs in general better than most known residual vector quantization schemes.

Recently, da Silva et al. [4] have developed a method to perform successive approximation vector quantization (SA-VQ) that does not need any codebook optimization. In it, the codebook $\mathcal{Y}_n$ of pass n is merely a scaled version of a mother codebook $\mathcal{Y}$. The codebook $\mathcal{Y}$ is such that all its vectors are on an hyper-sphere, and $\mathcal{Y}_n = \alpha^n \mathcal{Y}$, $0 < \alpha < 1$. In addition, $\mathcal{Y}$ can be based on regular lattices related to the solution of the sphere packing problem [5]. This implies that no codebook optimization is needed. An embedded wavelet coder similar to the one in [1], but replacing the successive approximation scalar quantization by the successive approximation vector quantization has been described in [4] (SA-W-VQ coder). It has been shown to perform consistently better than the scalar one, giving results comparable to the state-of-the-art in wavelet image coding. This implies that it performs at least equivalently to most known residual vector quantization schemes, but at a very small fraction of their complexity.

In other words, in SA-VQ, each vector is represented by a series of vectors of decreasing magnitudes (α^n) and orientations drawn from a fixed codebook $\mathcal{Y}$, referred to as the *orientation codebook* (all its vectors have the same magnitude and differ only in their orientations in k-dimensional space). More specifically, a k-dimensional vector $\mathbf{v}$ can be approximated by a vector $\mathbf{v}_P$ such that:

$$\mathbf{v}_P = \sum_{i=1}^{P} \alpha^i B \mathbf{u}_{n_i} \tag{3}$$

In eq. 3, α and B are fixed, $\mathbf{u}_{n_i} \in \mathcal{Y}$ and $\mathcal{Y}$ is such that if $\mathbf{u} \in \mathcal{Y}$ than $\|\mathbf{u}\| = 1$. In addition, $0 < \alpha < 1$ and B is a scalar chosen according to the dynamic range of the magnitudes of the vectors to be approximated.

A representation like the one in eq. 3 is useful for successive approximation coding only if $\exists$ B such that if $\|\mathbf{v}\| < M$, than $\mathbf{v}$ can be approximated with arbitrary precision by $\mathbf{v}_P$ if P is sufficiently large. In other words, $\mathbf{v}_P$ must *converge* to $\mathbf{v}$ as $P \to \infty$. However, for a given orientation codebook, it is not guaranteed that for every value of α the representations like the ones in eq. 3 are convergent. In [4], conditions to guarantee their convergence for worst case conditions have been derived on an experimental basis. For a given orientation codebook $\mathcal{Y}$, $\theta_{\max}$ was defined as the maximum possible angle between any vector $\mathbf{v} \in \mathbb{R}^k$ and its nearest neighbour in $\mathcal{Y}$. More formally, it is given by:

$$\theta_{\max} = \max_{\|\mathbf{v}\|=1} \{ \min_{\mathbf{u}_i \in \mathcal{Y}} [\arccos(\mathbf{v} \cdot \mathbf{u}_i)] \} \tag{4}$$

By worst case conditions it is meant that, $\forall n$, the angle between $\mathbf{v} - \mathbf{v}_n$ and $\mathbf{u}_{n+1}$ is equal to $\theta_{\max}$.

In the remaining of this paper, we first give an algorithm to compute a successive approximation decomposition of a vector according to eq. 3. Next, we state a theorem relating the values of α and $\theta_{\max}$ necessary for the convergence of SA-VQ, which confirms the worst case conditions determined experimentally in [4]. However, the simulation results of coding wavelet transforms of images presented in [4] show that these values of α are very conservative, and the best performance of this algorithm is obtained for values of α much smaller than the minimum ones mandated by the theorem. It is thus very desirable to have more realistic results related to the performance of the algorithm. Therefore, we next state a theorem which establishes that, after small modifications, SA-VQ is guaranteed to converge for these smaller values of α. Then, the image coding results of an SA-W-VQ coder incorporating these modifications are presented and discussed, followed by conclusions and suggestions for further improvements.

2. The SA-VQ algorithm

Given scalars B and α, and an orientation codebook $\mathcal{Y} = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N\}$, an algorithm for finding the representation in eq. 3 for a vector $\mathbf{v}$ is as follows:

1. Make $l = 0$ and $\mathbf{r}_0 = \mathbf{v}$.
2. Chose $n_l \in \{1, 2, \ldots, N\}$ such that
$$\mathbf{r}_l \cdot \mathbf{u}_{n_l} = \min_{1 \le j \le N} \{\mathbf{r}_l \cdot \mathbf{u}_j\}$$
3. Make $\mathbf{r}_{l+1} = \mathbf{r}_l - \alpha^{l+1} B \mathbf{u}_{n_l}$.
4. Increment l.
5. Return to step 2.

The algorithm stops either when the desired precision in the representation of $\mathbf{v}$ is obtained or when a maximum number of passes (bit-rate) is reached.

3. SA-VQ convergence

Convergence of the SA-VQ algorithm is equivalent to, in the algorithm described in section 2, $\|\mathbf{r}_l\|$ tending to zero as l tends to infinity. Supposing that $\|\mathbf{v}\| \leq B$, and that for every l the angle between $\mathbf{r}_l$ and $\mathbf{u}_{n_l}$ is equal to $\theta_{\max}$ as defined by eq. 4, the following theorem can be proved [6]:

Theorem 1 *The SA-VQ algorithm converges if:*

$$\alpha \geq \frac{1}{2\cos(\theta_{max})}, \quad \theta_{max} \leq 45^\circ \tag{5}$$

$$\alpha \geq \sin(\theta_{max}), \quad \theta_{max} \geq 45^\circ \tag{6}$$

The above equations match perfectly the graph given in [4], which has been determined experimentally.

The above equations can be conveniently interpreted considering eq. 3 and the algorithm in section 2. It can be shown that if the SA-VQ algorithm converges, then the residual after P passes, $\mathbf{r}_P = \mathbf{v} - \mathbf{v}_P$ is such that $\|\mathbf{r}_P\| \leq \alpha^P B$. This means that the smaller the α, the smaller the error after P passes. Since the more passes used to represent a vector, the larger the bit-rate, one could conclude from Theorem 1 that given an orientation codebook, one should use the smallest value of α possible if we want the minimum distortion for a given bit-rate. However, Theorem 1 gives the minimum values of α for the worst case conditions, where the orientation error in each pass is equal to the maximum possible. These conditions are therefore very conservative, and in practice one can have convergence for values of α much smaller than the ones given by Theorem 1. In fact, in the coder described in [4], the values of α which give peak PSNR performance are well below the ones from Theorem 1 (see the results for the "conventional" algorithm in figure 1).

It would be thus highly desirable to have some results concerning the average performance of SA-VQ, which could give more realistic values of α than the worst case ones in Theorem 1. This issue is addressed in the next section.

4. A modified version of SA-VQ with improved convergence

In order to an SA-W-VQ-like coder to converge for smaller values of α, the successive approximation vector quantization algorithm has to be modified to guarantee that $\alpha^{n+1} B \leq \|\mathbf{r}_n\| \leq \alpha^n B$ at every step. In order to achieve this, the zero vector, referred to as $\mathbf{u}_0$, has to be added to the codebook $\mathcal{Y}$, and therefore $\mathcal{Y} = \{\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_N\}$. The algorithm is then as follows:

1. Make $l = 0$, $t = 0$ and $\mathbf{r}_0 = \mathbf{v}$.
2. Chose $n_l \in \{1, 2, \ldots, N\}$ such that
$$\mathbf{r}_l \cdot \mathbf{u}_{n_l} = \min_{1 \leq j \leq N} \{\mathbf{r}_l \cdot \mathbf{u}_j\}$$
3. If $\|\mathbf{r}_l\| < \alpha^{l-t+1} B$ then Make $n_l = 0$.
4. If $\|\mathbf{r}_l\| > \alpha^{l-t} B$ then Increment t.
5. Make $\mathbf{r}_{l+1} = \mathbf{r}_l - \alpha^{l-t+1} B \mathbf{u}_{n_l}$.
6. Increment l.
7. Return to step 2.

It is important to note that in order for the decoder to keep track of the value of t for that vector, an escape code should be transmitted every time that t is incremented. This algorithm then guarantees that $\alpha^{n+1} B \leq \|\mathbf{r}_n\| \leq \alpha^n B$ at every step.

For this modified SA-VQ algorithm, we have the following theorem [6]:

Theorem 2 *Suppose that the orientation codebook used in SA-VQ has $\theta_{max} \leq 60^\circ$. Then the modified version of the SA-VQ algorithm converges for every $0 < \alpha < 1$.*

It is important to note that, as long as the value of $(\cos\theta)_{\max} \leq 60^\circ$, the modified SA-VQ algorithm will converge for $0 < \alpha < 1$ irrespective of the particular orientation codebook used. This result is then much stronger than the one of Theorem 1.

The compromises involving the choice of α according to Theorem 2 are somewhat different than the ones according to Theorem 1. In the algorithm leading to Theorem 1, the value of α should be as small as possible, in order for the residual to decrease as fast as possible. On the other hand, for the modified SA-VQ algorithm, leading to Theorem 2, if we make the value of α too small, than $\|\mathbf{r}_n\| > \alpha^n B$ with larger probability, and therefore, the final value of t will be larger, which implies that the residual will be larger. There is then a different compromise in choosing α. In the next section, we will present simulation results of a modified version of the SA-W-VQ algorithm from [4] incorporating the modifications in the SA-VQ algorithm leading to Theorem 2. These compromises will be then analyzed more closely.

5. Experimental results

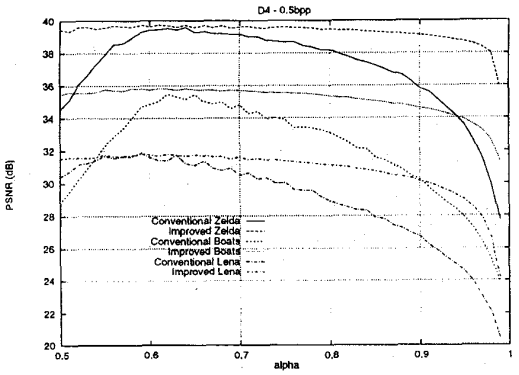
The images LENA 256×256, ZELDA and BOATS have been coded at 0.5bit/pixel using the "conventional" SA-W-VQ algorithm [4] as well as the "improved" SA-W-VQ, incorporating the modifications proposed in the previous section. The PSNRs of these images were plotted against values of α in the range $[0.50, 0.99]$. The results are shown in figures 1.a to 1.c for the first shell of the D_4, E_8 and Λ_{16} lattices [5] as orientation codebooks, respectively.

Table 1 shows the worst case values of α (eqs. 5 and 6) and the ranges of α which give the peak performances of the "conventional" and "improved" SA-W-VQ algorithms. The PSNR provided is the minimum value for the given ranges.

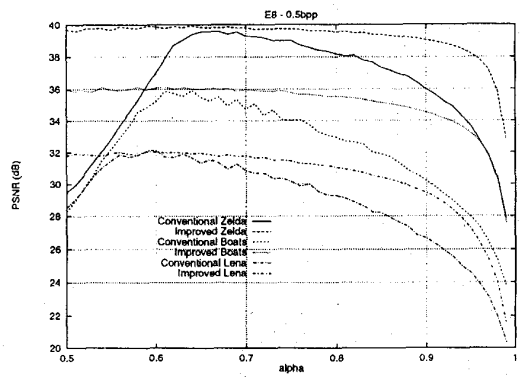
For the E_8 orientation codebook, $\cos(\theta_{\max}) = 0.71$. According to Theorem 1, this requires that $\alpha \geq 0.71$ in order to guarantee convergence. From this graph, it can be clearly observed that this value of α is very pessimistic. In the "old" SA-W-VQ coder, the value of α which gives the peak performance varies reasonably from image to image. For ZELDA the peak performance is obtained with $\alpha = 0.67$, for BOATS with $\alpha = 0.61$ and for LENA with α in the range $[0.58, 0.60]$. Therefore, in the "old" SA-W-VQ coder there was no single value of α that could be universally used for all images. Similar conclusions can be taken for the D_4 and Λ_{16} orientation codebooks.

In contrast, looking at the performance of the "improved" SA-W-VQ coder, it can be seen that the PSNR performance is almost independent of the value of α chosen. In addition, in some cases it gives even higher peak PSNR performance than the "old" one. This is a very desirable result, for we now have an algorithm whose performance does not depend too much on α, and therefore is almost image independent. In other words, one can safely choose α in the range $[0.5, 0.6]$ and guarantee a performance very close to the optimal.

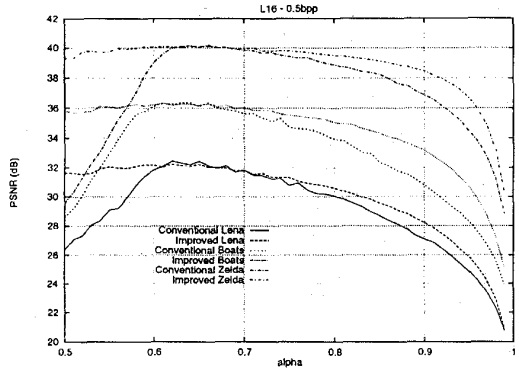
In figures 1.a to 1.c the PSNR was plotted only for $\alpha \in [0.5, 1)$, despite the fact that Theorem 3 guarantees convergence of the modified SA-VQ algorithm for $\alpha \in (0, 1)$. As mentioned in the previous section, there is a compromise in the choice of α, that is, we cannot chose α too small, because then the number of passes would increase (large values of t in the modified SA-VQ algorithm). This is confirmed in figure 3, where the PSNR of the "conventional" and "improved" SA-W-VQ algorithms is shown for $\alpha \in (0, 1)$ using the E_8 orientation codebook. There, it can be seen



(a)



(b)



(c)

Figure 1: α×PSNR for the images ZELDA, BOATS and LENA, using the "conventional" and "improved" SA-W-VQ algorithm, having as orientation codebook the lattice: a) D_4; b) E_8; c) Λ_{16}.

(a) (b)

Figure 2: a) The original LENA 256×256; b) Lena 256×256 coded with the improved SA-W-VQ algorithm at 0.4bit/pixel using Λ_{16} as the orientation codebook with $\alpha = 0.62$.

Table 1: Values of α for the worst case (eqs. 5 and 6), and ranges of α for peak performances of the conventional and "improved" SA-W-VQ algorithms, along with the minimum PSNR for the given ranges.

Image/ Orientation Codebook	α worst case	α peak performance conventional	α peak performance improved	PSNR minimum peak (dB)
LENA/D_4	0.71	[0.54,0.60]	[0.5,0.68]	31.51
BOATS/D_4	0.71	[0.62,0.62]	[0.5,0.78]	35.46
ZELDA/D_4	0.71	[0.60,0.64]	[0.5,0.83]	39.36
LENA/E_8	0.71	[0.58,0.60]	[0.5,0.66]	31.79
BOATS/E_8	0.71	[0.61,0.61]	[0.5,0.74]	35.83
ZELDA/E_8	0.71	[0.67,0.67]	[0.5,0.79]	39.60
LENA/Λ_{16}	0.82	[0.60,0.70]	[0.5,0.70]	31.59
BOATS/Λ_{16}	0.82	[0.59,0.69]	[0.5,0.73]	35.68
ZELDA/Λ_{16}	0.82	[0.61,0.77]	[0.5,0.82]	39.29

that for $\alpha < 0.5$ the performance decreases dramatically for both algorithms. However, even in this case the "improved" SA-W-VQ algorithm performs much better than the "conventional" one.

Finally, figure 2 shows the original LENA 256×256 image and the same image coded with the improved SA-W-VQ algorithm at 0.4bit/pixel using the Λ_{16} lattice as orientation codebook.

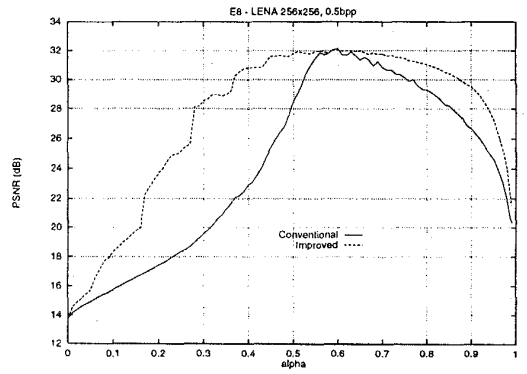


Figure 3: α×PSNR for the image LENA, using the "conventional" and "improved" SA-W-VQ algorithm, having as orientation codebook the E_8 lattice. α varies in the interval $(0, 1)$.

6. Conclusions

Summarizing, it can be said that the proposed modifications on the SA-VQ algorithm have definitely improved the SA-W-VQ coder. The analytical tools developed and the nice convergence behaviour shown open many new possibilities. In addition, the fact that the performance of the modified algorithm is almost independent of the values of α provided that α is large enough, makes the algorithm more suitable for applications where an optimization of the value of α would lead to prohibitive computational costs. A good example of this case can be found in video coding, where it would not be practical to optimize α from frame to frame.

It is interesting to note that in the same way that eq. 1 suggests that successive approximation scalar quantization, when $\alpha = 1/2$, is essentially equivalent to the representation of a real number in binary notation, one can argue that successive approximation vector quantization would be equivalent to a "binary" representation of a vector (see eq. 3). More specifically, one could say that it would be equivalent to the representation of a vector $\in \mathbb{R}^N$ in a "base" $1/\alpha$, given an orientation codebook (note that in the scalar case the orientation codebook is $\{-1, 0, 1\}$). This interpretation opens new theoretical and practical avenues. For example, this representation of a vector in a base $1/\alpha$ can be regarded as a "vector bit-plane decomposition". Therefore, SA-VQ can be easily used in applications that use bit-plane encoding, as in [7]. Indeed, many of the proposals to the standard JPEG 2000 use some form of bit-plane encoding.

Considering that the SA-W-VQ coder already gives rate×distortion performances comparable to the state-of-the-art, one can say that SA-W-VQ-like encoders are very promising.

7. 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] C. F. Barnes, S. A. Riziv, and N. M. Nasrabadi, "Advances in residual vector quantization: A review," *IEEE Transactions on Image Processing, Special Issue on Vector Quantization*, vol. 5, pp. 226–225, February 1996.

[4] 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.

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

[6] M. Craizer, E. A. B. da Silva, and E. G. Ramos, "New results on successive approximation vector quantization," *Electronics Letters*, vol. 34, January 1998.

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