Convergent algorithms for successive approximation vector quantisation with applications to wavelet image compression

M. Craizer, E.A.B. da Silva and E.G. Ramos

Abstract: Embedded wavelet coders have become very popular in image compression applications, owing to their simplicity and high coding efficiency. Most of them incorporate some form of successive approximation scalar quantisation. Recently developed algorithms for successive approximation vector quantisation have been shown to be capable of outperforming successive approximation scalar quantisation ones. In the paper, some algorithms for successive approximation vector quantisation are analysed. Results that were previously known only on an experimental basis are derived analytically. An improved algorithm is also developed and is proved to be convergent. These algorithms are applied to the coding of wavelet coefficients of images. Experimental results show that the improved algorithm is more stable in a rate × distortion sense, while maintaining coding performances compatible with the state-of-the-art.

1 Introduction

Wavelet transform-based image coders are among the state of the art in image compression [1–4]. The wavelet transform of an image is a multi-resolution representation of it, where vertical, horizontal and diagonal image detail are represented in different resolutions [5].

One favourable property of wavelet transforms is that their coefficients in different resolutions and the same orientation exhibit strong similarities. This fact leads to the idea of zerotrees, initially proposed in [6] and used very efficiently in [1]. The above mentioned similarity among wavelet coefficients in different resolutions and the same orientation is in the following sense: if a wavelet coefficient is small, then the coefficients of the same spatial orientation and higher frequency have a great probability of also being small. When these coefficients are quantised to zero, their positions form a zerotree. In other words, a large number of zero coefficients can be represented with a single zerotree symbol. As this is a highly likely event, it leads to good coding efficiency.

When it comes to quantising and coding the wavelet coefficients, the methods that use successive approximations of the wavelet coefficients are among the ones having the best coding performances [1–3]. The successive approximation processes guarantee that the quantisation distortion of each coefficient does not exceed a certain maximum, and that all the bands tend to have the same level of distortion. In [1] and [2], successive approximation scalar quantisation (SASQ) for the non-zero coefficients is used. In [7], da Silva *et al.* used successive approximation vector quantisation (SAVQ) for the non-zero coefficients. In their experiments, SAVQ performed better than SASQ in quantising and coding wavelet coefficients.

In this paper, we will state and prove some properties of the SAVQ algorithm, as well as propose and analyse a new version of it. This modified algorithm possesses more stable rate × distortion performance, in the sense that its parameters can be set such that the peak PSNR performance is obtained irrespective of the particular image coded. This is a very desirable result, as finding the optimum parameters for each input image would be computationally expensive.

2 Successive approximation vector quantisation

In SAVQ, an orientation codebook $C = \{v_1, v_2, \ldots, v_q\}$ consisting of unitary vectors from R^N is considered. The idea of the algorithm is to approximate a vector $x \in R^N$, $\|x\| \leq X_{\max}$, by a linear combination of the form

$$x_n = \sum_{j=1}^{n} \alpha^j X_{\max} v_{i_j} \tag{1}$$

where $v_{i_j} \in C$, and $0 < \alpha < 1$ is a parameter of the algorithm. The indexes $(i_1, i_2, \ldots)$ represent the vector x. They are characterised by the fact that each v_{i_j} is the codebook vector nearest to r_{j-1}, where $r_n = x - x_n$ is the residual vector.

The algorithm to obtain the sequence $(i_1, i_2, \ldots)$ is as follows [7]:

Step 1: Start with $w = x$ and $n = 1$.
Step 2: Choose $i_n \in \{1, \ldots, q\}$, such that

$$w \cdot v_{i_n} = \max_{1 \leq k \leq q} \{w \cdot v_k\}$$

Step 3: Replace the vector w by $w - \alpha^n X_{\max} v_{i_n}$.
Step 4: Increment n by 1.
Step 5: If $\|w\| > T$, where T is a predetermined threshold, return to step 2.
Step 6. Stop.

IEE Proceedings online no. 19990022
DOI: 10.1049/ip-vis:19990022
Paper first received 26th June and in revised form 29th October 1998
M. Craizer is with the Departamento de Matemática/PUC-Rio, Rua Marquês de São Vicente, 225, Rio de Janeiro, RJ 22453-900, Brazil
E.A.B. da Silva and E.G. Ramos are with Universidade Federal do Rio de Janeiro, Cx. P. 68504, Rio de Janeiro, RJ 21945-970, Brazil

The algorithm is said to be convergent if, for any vector $x \in R^N$, $\|x\| \leq X_{max}$, the residual vector tends to zero, when n goes to infinity.

Some important observations can be made about the SAVQ algorithm. The first is that it has many similarities to Mallat's matching pursuit algorithm [9]. The main difference is that, in Mallat's matching pursuits, we replace the $\alpha^j X_{max}$ term in eqn. 1 with the projection of r_{j-1} on the subspace generated by v_{i_j}. More precisely, let $\gamma_j = r_{j-1} \cdot v_{i_j}$, where v_{i_j} is the codebook vector nearest to r_{j-1}, and the dot $(\cdot)$ is denoting the inner product between vectors. The matching pursuit algorithm approximates a vector x by a vector of the form

$$x_n = \sum_{j=1}^{n} \gamma_j v_{i_j} \qquad (2)$$

This implies that, in Mallat's matching pursuits, a vector is represented by a sequence of unit vectors $v_{i1}, v_{i_2}, \ldots$, plus a sequence of scalars $\gamma_1, \gamma_2, \ldots$, whereas, in eqn. 1, a vector is represented by just a sequence of unit vectors $v_{i_1}, v_{i_2}, \ldots$. In coding applications, this may represent an advantage.

At with Mallat's matching pursuits, the SAVQ algorithm determines, in each pass, the vector $v_{i_{j+1}}$ that is closest to the residual r_j, and therefore minimises the error in the representation of r_j in that pass. However, this procedure is not guaranteed to generate the optimum representation, that is, the one that yields the minimum representation error after n passes. More precisely, if the algorithm above generates a sequence of vectors $v_{i_1}, v_{i_2}, \ldots, v_{i_n}$, that, according to eqn. 1, provides an approximation x_n of x, there is no guarantee that there is not a different sequence of vectors $v_{j_1}, v_{j_2}, \ldots, v_{j_n}$ providing an approximation x'_n to x such that $\|x - x'_n\| < \|x - x_n\|$. In other words, this discussion implies that, besides the fact that a representation as in eqn. 1 is not unique, the above algorithm will not necessarily find the optimum one. Fortunately, in most cases, the approximation it finds performs well enough.

An important parameter associated with $\mathcal{C}$ is the parameter θ_{max}, which is the maximum value of the angle between a vector in R^N and the codebook vector nearest to it. More precisely,

$$\theta_{max} = \max_{\substack{x \in R^N \\ \|x\|=1}} \{\min_{v_i \in \mathcal{C}} \{\arccos(x \cdot v_i)\}\}$$

In [7], the behaviour of the algorithm for various values of α and θ_{max} was simulated for the worst case, i.e. when, in every pass, the angular error is equal to θ_{max}. It has been observed that, for every θ_{max}, there is a value $\bar{\alpha}$ such that, if $\alpha \geq \bar{\alpha}$, the algorithm converges. In this paper, we prove the following theorem (which had been stated without proof in [8]):

Theorem 1: The SAVQ algorithm converges if

$$\alpha \geq \frac{1}{2\cos(\theta_{max})}, \ \theta_{max} \leq \frac{\pi}{4} \qquad (3)$$

$$\alpha \geq \sin(\theta_{max}), \ \theta_{max} \geq \frac{\pi}{4} \qquad (4)$$

The proof of this theorem is given in the Appendix (Section 8.1).

The above theorem gives us some insight into the choice of the orientation codebook C. As will be seen in the following Section, the error $\|x - x_n\|$ is bounded by a quantity proportional to α^n. Therefore the smaller the value of α, the smaller the error in the representation of eqn. 1 when just n terms are used. By the above theorem, we must have small θ_{max} to have small $\bar{\alpha}$. Thus, it is important that the orientation codebook has a θ_{max} as small as possible without increasing the number of vectors in it, as the more vectors that are used, the more bits are necessary to represent them. This suggests that the codebook C be such that its vectors are as 'uniformly' distributed as possible on the N-dimensional hyper-sphere with unitary radius. Therefore codebooks related to the sphere packing problem [10] were chosen in the simulations of Section 4.

However, simulation results such as the ones presented in [7] show that the choice of α based on eqns. 3 and 4 is very conservative. The algorithm gives better results with the parameter α smaller than the ones above. In fact, the average of the angles θ_j between a residual $r_j, j = 0, 1, \ldots,$ and its nearest codebook vector is much smaller than θ_{max}. This suggested to us the study of the SAVQ algorithm with smaller values of α. This is the subject of the following Section.

3 Modified SAVQ algorithm

In this Section, we describe the improved version of the SAVQ algorithm. As we mentioned in the introduction, this version is convergent for a wider range of α values, and possesses a peak PSNR performance much more independent of the specific value of α used.

In order for the SA-VQ algorithm to converge for a wider range of α values, it has to be modified to guarantee that $\alpha^{n+1} X_{max} \leq \|r_n\| \leq \alpha^n X_{max}$ at each step. This is done as follows: If the magnitude of r_n is smaller than $\alpha^{n+1} X_{max}$, the zero vector is transmitted, so that there is no refinement for that vector in that pass. If the magnitude of r_n is greater than $\alpha^n X_{max}$, an escape code is transmitted, so that both coder and decoder know that the exponent of α should not be incremented for that vector in that pass.

More precisely, we shall represent a vector x by a sequence $(i_{j,k})$, $0 \leq i_{j,k} \leq q$, such that x can be approximated by

$$x_{n,l} = \sum_{j=1}^{n} \alpha^j X_{max} \sum_{k=1}^{l} v_{i_{j,k}} \qquad (5)$$

where $v_{i_{j,k}} \in \mathcal{C}$. It is important to mention that, for this version of the SAVQ algorithm, the zero vector is included in the codebook $\mathcal{C}$.

The sequence $(i_{n,l})$, $0 \leq i_{n,l} \leq q$, is obtained by the following algorithm:

Step 1: Start with $w = x$, $n = 1$ and $l = 1$.
Step 2: If $\|w\| \leq \alpha^n X_{max}$, make $i_{n,l} = 0$.
Step 3: If $\|w\| > \alpha^n X_{max}$, choose $i_{n,l} \in \{1, \ldots, q\}$ such that

$$w \cdot v_{i_{n,l}} = \max_{1 \leq k \leq q} \{w \cdot v_k\}$$

Step 4: Replace w by $w - \alpha^n X_{max} v_{i_{n,l}}$.
Step 5: If $\|w\| > \alpha^n X_{max}$, increment l by 1 and return to step 3.
Step 6: Increment n by 1 and make $l = 1$.
Step 7: If $\|w\| > T$, where T is a predetermined threshold, return to step 1.
Step 8: Stop.

We shall let $r_{n,l} = r_{n,l}(x) = x - x_{n,l}$. It has the property that

$$\alpha^{n+1} X_{max} \leq \|r_{n,l}(x)\| \leq \alpha^n X_{max}$$

for each $n \in N$, and $1 \leq l \leq L(n) = L(n, x)$. Hence, if $L(n, x)$ is bounded, it is clear that $x_{n,l}$ converges to x. Therefore we say that the modified SAVQ algorithm is convergent for a particular choice of α, if there exists L, independent of x and n, such that $L(n, x) \leq L$, for $\|x\| \leq X_{max}$.

We can prove the following theorem:

Theorem 2: Suppose that the orientation codebook used in the SAVQ algorithm has $\theta_{max} \leq \pi/3$. Then, the modified version of the SAVQ algorithm converges for $0 < \alpha < 1$.

This theorem will be proved in the Appendix (Section 8.2).

This theorem states that the modified SAVQ algorithm is convergent for any $0 < \alpha < 1$. However, for the algorithm to be practical, it is important that $L(n, x)$ is small, with great probability. For this reason, we used only $0.5 \leq \alpha < 1$ in our experiments. We observed in these experiments that $L(n, x) = 1$ occurred with a probability near to 1. Also, as will be seen in Section 4, the experimental results show that the modified SAVQ algorithm is more stable with respect to the choice of α. This is entirely in accord with theorem 2.

4 Experimental results

The images 'Lena' 512×512, 'Zelda' and 'Boats' have been coded at 0.5 bit/pixel, using the 'conventional' SAWVQ algorithm [7], as well as the 'improved' SAWVQ algorithm with the modifications proposed in the preceding Section. The PSNR of these images was plotted against values of alpha in the range [0.50, 0.99]. The results are shown in Fig. 1–3 for the first shell of the D_4, E_8 and Λ_{16} lattices [10] as orientation codebooks, respectively. For D_4, we use 2×2 blocks in all bands; for E_8, we use either 2×4 or 4×2 blocks; for Λ_{16}, we use 4×4 blocks.

The values of θ_{max} for the first shells of lattices D_4, E_8 and Λ_{16} are listed in Table 1. The worst case values of α (eqns. 3 and 4) are also shown.

Table 2 shows, together with the worst case values of α from Table 1, the ranges of α that give the best performances of the 'conventional' and 'improved' SAWVQ 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 eqns. 3 and 4, this requires that $\alpha \geq 0.71$ to guarantee convergence. From Fig. 2, it can be clearly observed that this value of α is very pessimistic. In the 'conventional' SAWVQ coder, the value of α that gives the peak performance varies reasonably from image to image. For 'Zelda', the peak performance is obtained with $\alpha = 0.67$, for 'Boats' it is obtained with $\alpha = 0.61$, and for 'Lena' it is obtained with α in the range [0.57, 0.62]. Therefore, in the 'conventional' SAWVQ coder, there was no single value of α that could be universally used for all images. Similar conclusions can be drawn for the D_4 and Λ_{16} orientation codebooks.

In contrast, looking at the performance of the 'improved' SAWVQ coder, it can be seen that the

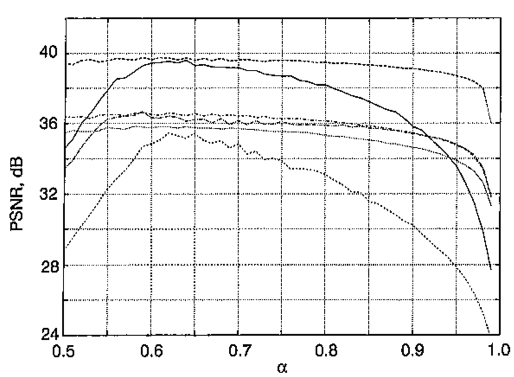

Fig. 1 *$\alpha \times$ PSNR for 'Zelda', 'Boats' and 'Lena' 512×512 images using conventional and improved SAWVQ algorithm with D_4 lattice as orientation codebook*

——— conventional Zelda
– – – – improved Zelda
- - - - - conventional Boats
.... improved Boats
—·—·— conventional Lena
–·–·–·– improved Lena

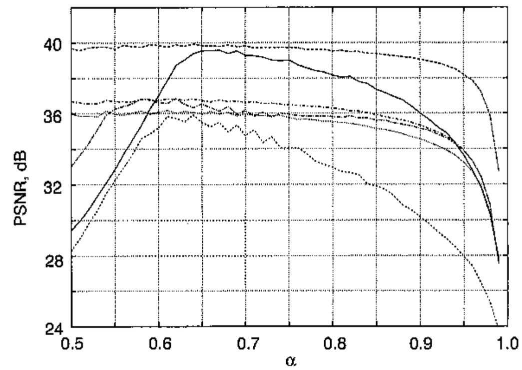

Fig. 2 *$\alpha \times$ PSNR for 'Zelda', 'Boats' and 'Lena' 512×512 images using conventional and improved SAWVQ algorithm with E_8 lattice as orientation codebook*

——— conventional Zelda
– – – – improved Zelda
- - - - - conventional Boats
.... improved Boats
—·—·— conventional Lena
–·–·–·– improved Lena

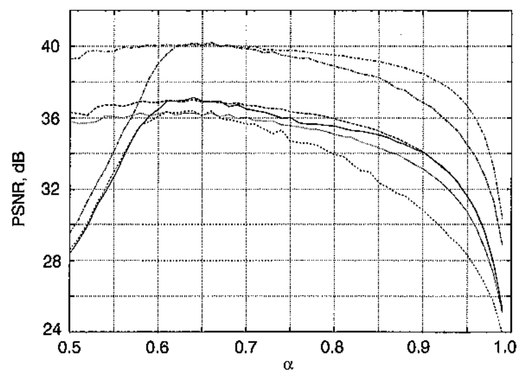

Fig. 3 *$\alpha \times$ PSNR for 'Zelda', 'Boats' and 'Lena' 512×512 images using conventional and improved SAWVQ algorithm with λ_{16} lattice as orientation codebook*

——— conventional Zelda
– – – – improved Zelda
- - - - - conventional Boats
.... improved Boats
—·—·— conventional Lena
–·–·–·– improved Lena

Table 1: Values of θ_{max} for first shells of lattices D_4, E_8 and Λ_{16}, with minimum values of α according to eqns. 3 and 4

	D_4	E_8	Λ_{16}
θ_{max}	45°	45°	55°
α	0.71	0.71	0.82

Table 2: Values of α for worst case (eqns. 3 and 4), and ranges of α for best performances of conventional and 'improved' SAWVQ algorithms, with minimum PSNR for given ranges

Image/ orientation codebook	α worst case	α best performance, conventional	α best performance, improved	Minimum PSNR for range, dB
Lena/D_4	0.71	[0.56,0.63]	[0.5,0.74]	36.30
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	[0.57,0.62]	[0.5,0.73]	36.54
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	[0.60,0.73]	[0.5,0.78]	36.15
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

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 'conventional' one. This is a very desirable result, for we now have an algorithm whose performance does not depend too much on α. In fact, we observe that we can safely choose any α in the range [0.5, 0.65] and still guarantee a performance very close to the optimum.

In other words, we can apply the improved SAWVQ with a fixed value of the parameter α and still obtain practically peak performance for any image. This prevents us from carrying out a computationally complex optimisation of the value of α for every input image. Therefore the improved SAWVQ algorithm has a chief advantage over the one in [7] in applications where implementation complexity is an important concern.

In Figs. 1–3, the PSNR was plotted only for $\alpha \in [0.5, 1)$, despite the fact that theorem 3 guarantees convergence of the modified SAVQ algorithm for $\alpha \in (0, 1)$. As stated in Section 3, only values of $\alpha \in [0.5, 1)$ would give practical algorithms. This is confirmed in Fig. 4, where the PSNR, for Lena 256×256 at 0.5 bit/pixel, of the 'conventional' and 'improved' SAWVQ algorithms is shown for $\alpha \in (0, 1)$ using the E_8 orientation codebook. There, it can be seen that for $\alpha < 0.5$, the performance decreases dramatically for both algorithms. However, even in this case, the 'improved' SAWVQ algorithm performs much better than the 'conventional' one.

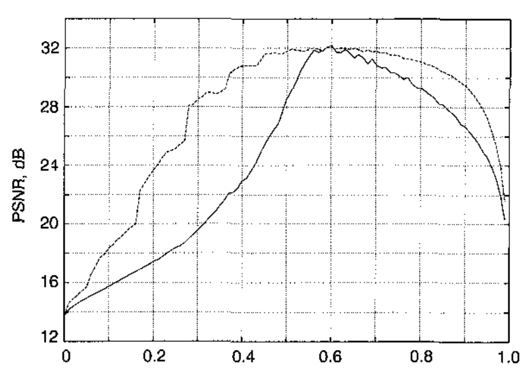

Fig. 4 $\alpha \times$ *PSNR for image 'Lena', using conventional and improved SAWVQ algorithm, with E_8 lattice as orientation codebook*
α varies in interval (0,1)
——— conventional
– – – – improved

a

b

Fig. 5 *'Lena' 256×256 image*
a Original
b Coded with improved SAWVQ algorithm at 0.4 bit/pixel, using Λ_{16} as orientation codebook $\alpha = 0.62$

Finally, Figs. 5 show the original Lena 256×256 image and the same image coded with the improved SAWVQ algorithm at 0.4bit/pixel, using the Λ_{16} lattice as orientation codebook.

5 Conclusions

In this paper, we have analysed some properties of successive approximation vector quantisation (SAVQ). Aside from proving the convergence of the SAVQ algorithm used in [7], we proposed an improved version of it. The improved algorithm has better convergence properties and definitely yields better experimental results. Considering that the SAWVQ coder already gives rate $\times$ distortion performances comparable with the state of the art (e.g. 37.14 dB for Lena 512×512 at 0.5 bit/pixel), these are very promising results.

Among areas for future investigation is the search for codebooks with better rate $\times$ distortion properties than the lattices associated with the sphere packing problem. Candidates for such codebooks are, for example, the ones in [11] that describe different ways to place an arbitrary number of points 'uniformly' on an N-dimensional hypersphere, for

$N=3, 4, 5$. LBG-type VQ algorithms for the generation of this kind of codebook can also be considered.

Also, the good convergence behaviour shown by the SAVQ algorithm opens many new possibilities. For example, as in the analysis carried out in this paper we have not restricted the dimension N to being finite, the successive approximation vector quantisation theory can be, in principle, generalised to Hilbert spaces. This is an exciting possibility, because it can lead to the development of different classes of signal decomposition, opening new avenues for research in signal compression.

6 Acknowledgments

This work has been supported in part by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) and the programs RECOPE and PRONEX from the Brazilian Government.

7 References

1 SHAPIRO, J.M.: 'Embedded image coding using zerotrees of wavelet coefficients', *IEEE Trans. Acoust. Speech Signal Process.*, 1993, **41**, pp. 3445–3462
2 SAID, A., and PEARLMAN, W.A.: 'A new, fast and efficient image codec based on set partitioning in hierarchical trees', *IEEE Trans. Circuits Syst. Video Technol.*, 1996, **6**, pp. 243–250
3 PEARLMAN, W.A., and SAID, A.: 'A survey of the state-of-the-art and utilization of embedded, tree-based coding'. Proceedings of the 1998 IEEE Int. Symposium on Circuits and systems, Monterey, California, May 1998
4 ALGAZI, R.V., and ESTES, R.E.: Jr., 'Analysis-based coding of image transforms and subband coefficients', *Proc. SPIE-Int. Soc. Opt Eng.*, 1995, **2564**, pp. 11–21
5 VETTERLI, M., and KOVAČEVIĆ, J.: 'Wavelets and subband coding' (Prentice Hall PTR, Englewood Cliffs, New Jersey, 1995)
6 LEWIS, A.S., and KNOWLES, G.: 'Image compression using the 2-D wavelet transform', *IEEE Trans. Image Process.*, 1992, **1**, pp. 244–250
7 DA SILVA, E.A.B., SAMPSON, D.G., and GHANBARI, M.: 'A successive approximation vector quantizer for wavelet transform image coding', *IEEE Trans. Image Process.*, (Special Issue on Vector quantization, February 1996, **5**, pp. 299–310
8 CRAIZER, M., DA SILVA, E.A.B., and RAMOS, E.G.: 'New results on successive approximation vector quantization', *Electron. Lett.*, 1998, **34**, pp. 59–60
9 MALLAT, S.G., and ZHANG, Z.: 'Matching pursuits with time-frequency dictionaes', *IEEE Trans. Signal Process.*, 1993, **41**, pp. 3397–3415
10 CONWAY, J.H., and SLOANE, N.J.A.: 'Sphere packings, lattices and groups' (Springer-Verlag, New York, 1988)
11 HARDIN, R.H., SLOANE, N.J.A., and SMITH, W.D.: 'Spherical codes'. URL: http://www.research.att.com/ njas/packings/index.html.

8 Appendix

8.1 Proof of theorem 1

We now prove theorem 1. In the following proposition, we give conditions on α for the SAVQ algorithm above to converge.

Proposition 1: If we choose $\alpha \geq 1/2\cos\theta_{\max}$, then $\|r_n\| \leq \alpha^n X_{\max}$

Proof: The formula

$$\|r_n\| \leq \alpha^n X_{\max} \tag{6}$$

will be proved by induction. It holds clearly for $n=0$. Assume that it holds also for $n=p$. We have from the cosine law that

$$\|r_{p+1}\|^2 = \|r_p\|^2 + \alpha^{2(p+1)}X_{\max}^2 - 2\|r_p\|\alpha^{p+1}X_{\max}\cos(\theta_p)$$

where θ_p is the angle between r_p and v_{i_p}. As $\theta_p \leq \theta_{\max}$,

$$\|r_{p+1}\|^2 \leq \|r_p\|^2 + \alpha^{2(p+1)}X_{\max}^2 - 2\|r_p\|\alpha^{p+1}X_{\max}\cos(\theta_{\max}) \tag{7}$$

The second member of this inequality is quadratic in $\|r_p\|$ and, hence, has its maximum value at one extreme of the interval $[0, \alpha^p X_{\max}]$. As, by hypothesis, $\alpha \geq 1/2\cos\theta_{\max}$, the maximum value is attained at $\|r_p\|=0$. Therefore

$$\|r_{p+1}\|^2 \leq \alpha^{2(p+1)}X_{\max}^2$$

which is equivalent to

$$\|r_{p+1}\| \leq \alpha^{p+1}X_{\max}$$

thus proving that expr. 6 holds for $n=p+1$. This completes the induction and proves the proposition.

If $\theta_{\max} \geq \pi/4$, we can choose smaller values of α. We have the following proposition:

Proposition 2: Assume that $\theta_{\max} \geq \pi/4$ and take $\alpha \geq \sin(\theta_{\max})$. Then $\|r_n\| \leq B\alpha^n$, where $B=\alpha X_{\max}/\cos(\theta_{\max})$.

Proof: The formula

$$\|r_n\| \leq B\alpha^n \tag{8}$$

will be also proved by induction. Observe first that, as $X_{\max} \leq B$, expr. 8 holds for $n=0$.

Assume that expn. 8 holds for $n=p$. We have from expr. 7 that

$$\|r_{p+1}\|^2 \leq \|r_p\|^2 + \alpha^{2(p+1)}X_{\max}^2 - 2\|r_p\|\alpha^{p+1}X_{\max}\cos(\theta_{\max})$$

where, by the induction hypothesis, $\|r_p\| \leq \alpha^p B$. The right member of this inequality is a quadratic function of $\|r_p\|$, and, hence, the maximum is achieved at one extreme of the interval $[0, \alpha^p B]$. However, the hypothesis $\theta_{\max} \geq \pi/4$ implies that, in fact, this maximum is achieved at $\|r_p\| = \alpha^p B$. Therefore

$$\|r_{p+1}\|^2 \leq \alpha^{2p}B^2\sin^2(\theta_{\max})$$

As $\alpha \geq \sin(\theta_{\max})$, we conclude that expr. 8 holds for $n=p+1$, thus completing the induction step.

The propositions 1 and 2 above easily imply theorem 1.

8.2 Proof of theorem 2

In this section, we prove theorem 2. It will be a consequence of some simple lemmas. We observe first that, from the description of the algorithm, it is clear that $\alpha^{n+1}X_{\max} \leq \|r_{n,l}(x)\| \leq \alpha^n X_{\max}$.

Lemma 1: Let $\theta_{n,l}$ be the angle between $r_{n,l}(x)$ and $v_{i_{n,l+1}}$. If $\cos(\theta_{n,l}) \geq 1/2\alpha$ (region (i) in Fig. 6) or else $\cos(\theta_{n,l}) < 1/2\alpha$ and $\|r_{n,l}(x)\| \leq 2\alpha^{n+1}X_{\max}\cos(\theta_{n,l})$ (region (ii) in Fig. 6), then $\|r_{n,l}(x) - \alpha^{n+1}X_{\max}v_{i_{n,l+1}}\| \leq \alpha^{n+1}X_{\max}$.

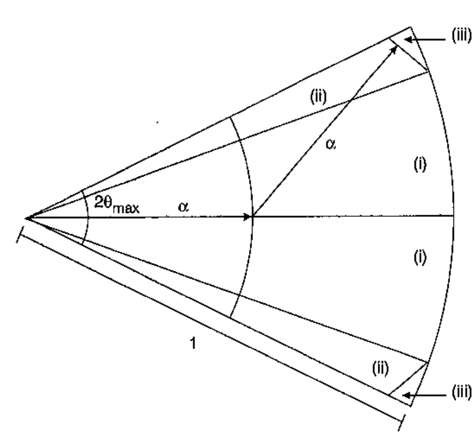

Fig. 6 *Voronoy cell of codebook vector in the two-dimensional case*

Proof: We have, by the cosine law, that

$$\|r_{n,l}(x) - \alpha^{n+1}X_{\max}v_{i_{n,l+1}}\|^2 = \|r_{n,l}(x)\|^2 + \alpha^{2(n+1)}X_{\max}^2 - 2\alpha^{n+1}X_{\max}\|r_{n,l}(x)\|\cos(\theta_{n,l})$$

Hence, if $\cos(\theta_{n,l}) \geq 1/2\alpha$,

$$\|r_{n,l}(x) - \alpha^{n+1}X_{\max}v_{i_{n,l+1}}\|^2 \leq \|r_{n,l}(x)\|^2 + \alpha^{2(n+1)}X_{\max}^2 - \alpha^n X_{\max}\|r_{n,l}(x)\|$$

and therefore

$$\|r_{n,l}(x) - \alpha^{n+1}X_{\max}v_{i_{n,l+1}}\| \leq \alpha^{n+1}X_{\max}$$

thus proving the first part of the lemma. Also, if $\|r_{n,l}(x)\| \leq 2\alpha^{n+1}X_{\max}\cos(\theta_{n,l})$, then clearly eqn. 9 implies that

$$\|r_{n,l}(x) - \alpha^{n+1}X_{\max}v_{i_{n,l+1}}\| \leq \alpha^{n+1}X_{\max}$$

thus completing the proof of the lemma.

Lemma 2: Assume that $\theta_{\max} \leq \pi/3$. There exists $\lambda = \lambda(\alpha, \theta_{\max}) < 1$ such that, if $\cos(\theta_{n,l}) \leq 1/2\alpha$ and $\|r_{n,l}(x)\| \geq 2\alpha^{n+1}X_{\max}\cos(\theta_{n,l})$ (see region (iii) in Fig. 6), then $\|r_{n,l}(x) - \alpha^{n+1}X_{\max}v_{i_{n,l+1}}\| \leq \lambda\|r_{n,l}(x)\|$.

Proof: Let

$$\lambda(\alpha, \theta_{\max}) = \max\left(\sqrt{1 + \alpha^2 - 2\alpha\cos(\theta_{\max})}, \frac{1}{2\cos(\theta_{\max})}\right)$$

As $\theta_{\max} < \pi/3$, $\lambda(\alpha, \theta_{\max}) < 1$. Now expr. 9 implies that

$$\left(\frac{\|r_{n,l}(x) - \alpha^{n+1}X_{\max}v_{i_{n,l+1}}\|}{\|r_{n,l}(x)\|}\right)^2 = 1 + \left(\frac{\alpha^{n+1}X_{\max}}{\|r_{n,l}(x)\|}\right)^2 - 2\frac{\alpha^{n+1}X_{\max}}{\|r_{n,l}(x)\|}\cos(\theta_{n,l})$$

with $\alpha \leq \alpha^{n+1}X_{\max}/\|r_{n,l}(x)\| \leq 1/2\cos(\theta_{n,l})$. As the second member is quadratic in $\alpha^{n+1}X_{\max}/\|r_{n,l}(x)\|$, we have

$$\left(\frac{\|r_{n,l}(x) - \alpha^{n+1}X_{\max}v_{i_{n,l+1}}\|}{\|r_{n,l}(x)\|}\right)^2 \leq \max\left(1 + \alpha^2 - 2\alpha\cos(\theta_{n,l}), \left(\frac{1}{2\cos\theta_{n,l}}\right)^2\right)$$

However, $\cos(\theta_{n,l}) \geq \cos(\theta_{\max})$ and therefore

$$\frac{\|r_{n,l}(x) - \alpha^{n+1}X_{\max}v_{i_{n,l+1}}\|}{\|r_{n,l}(x)\|} \leq \lambda(\alpha, \theta_{\max})$$

thus proving lemma 2.

We now complete the proof of the theorem.

Lemma 1 implies that, if $\cos(\theta_{n,l}) \geq 1/2\alpha$ or $\|r_{n,l}(x)\| \leq 2\alpha^{n+1}X_{\max}\cos(\theta_{n,l})$, then certainly $L(n, x) = l$. Lemma 2 shows that, if this is not the case,

$$\frac{\|r_{n,l}(x) - \alpha^{n+1}X_{\max}v_{i_{n,l+1}}\|}{\|r_{n,l}(x)\|} \leq \lambda$$

Therefore, after at most $\log\alpha/\log\lambda$ steps, we will have the occurrence of $\|r_{n,l}(x) - \alpha^{n+1}X_{\max}v_{i_{n,l+1}}\| \leq \alpha^{n+1}X_{\max}$. This shows that

$$L \leq \frac{\log\alpha}{\log\lambda}$$

thus proving the theorem.