Available online at www.sciencedirect.com
SCIENCE DIRECT®

Applied and Computational Harmonic Analysis

ACADEMIC PRESS

Appl. Comput. Harmon. Anal. 13 (2002) 103–115

www.academicpress.com

Alpha-expansions: a class of frame decompositions

M. Craizer,[a,*] D.A. Fonini Jr.,[b] and E.A.B. da Silva[b]

[a] R.Mq. de São Vicente 225, Gávea, Rio de Janeiro, RJ 22453-900, Brazil
[b] PEE/COPPE-UFRJ, Rio de Janeiro, RJ, Brazil

Received 16 January 2001; revised 7 November 2001

Communicated by S.G. Mallat

Abstract

This article analyzes a scheme for frame decompositions that is called α-expansion. In this scheme, the choice of a parameter α adequate to a given frame is a central point. We develop a theory that helps choosing the parameter α and also suggests algorithms for obtaining the α-expansions. The method is applied to frame expansions coding. In this context we give conditions under which, for high rate coding, α-expansions are better, in a rate $\times$ distortion sense, than schemes that find the frame coefficients first and quantize them in a second step.

Keywords: Frame decompositions; Successive approximations; Vector quantization; Matching pursuits

1. Introduction

In signal processing, the compact representation of overcomplete basis expansions, also called frame expansions, has been recognized as an important issue. For example, when we sample an analog signal above the Nyquist rate, the amplitudes are coefficients of a frame expansion. How to code these coefficients is a relevant question (see [14,15]). In this article, we shall analyze a class of frame expansions in the N-dimensional space. We refer to them as α-expansions, since they depend strongly on a parameter α in the interval $(0,1)$.

Instead of beginning with a frame, we shall consider the slightly more general notion of a codebook, which is simply a collection of vectors in $\mathbf{R}^N$. Consider a codebook $\mathcal{C}=\{\mathbf{v}_1,\ldots,\mathbf{v}_q\}$ and a real number α in the interval $(0,1)$. A representation of a vector $\mathbf{x}\in\mathbf{R}^N$ in the form

$$\mathbf{x}=\sum_{i=0}^{\infty}\alpha^i\mathbf{v}_{k_i}, \tag{1}$$

with $\mathbf{v}_{k_i}\in\mathcal{C}$, is called an α-expansion over $\mathcal{C}$, or simply an α-expansion. The problem we shall deal with in this article is to find the values of α such that the representation above holds for a large set of vectors $\mathbf{x}\in\mathbf{R}^N$.

The error of approximation of a vector $\mathbf{x}\in\mathbf{R}^N$ by a n term sum as above is given by

$$\mathbf{r}_n(\mathbf{x})=\mathbf{x}-\sum_{i=0}^{n-1}\alpha^i\mathbf{v}_{k_i}. \tag{2}$$

* Corresponding author.
E-mail addresses: craizer@mat.puc-rio.br (M. Craizer), fonini@lps.ufrj.br (D.A. Fonini), eduardo@lps.ufrj.br (E.A.B. da Silva).

1063-5203/02/$ – see front matter
PII: S1063-5203(02)00504-3

Denoting by $\|\cdot\|$ the Euclidean norm of a vector, we observe that $\|\mathbf{r}_{n+1}(\mathbf{x}) - \mathbf{r}_n(\mathbf{x})\|$ converges exponentially to zero with a rate α. Therefore it is important to choose the parameter α as small as possible. On the other hand, if α is small, the representation (1) will not hold for many vectors $\mathbf{x} \in \mathbf{R}^N$. Hence α must be large enough such that the α-expansion over $\mathcal{C}$ remains valid for a large set of vectors $\mathbf{x} \in \mathbf{R}^N$.

It is worthwhile to mention at this point a very well-known example of an α-expansion over a codebook. If we take the codebook to be

$$\mathcal{B}_N = \left\{\mathbf{v} = (\varepsilon_1, \ldots, \varepsilon_N) \;\middle|\; \varepsilon_i \in \{-1, 1\}, 1 \leqslant i \leqslant N\right\} \tag{3}$$

and the parameter $\alpha = \frac{1}{2}$, then the representation

$$\mathbf{x} = \sum_{i=0}^{\infty} \left(\frac{1}{2}\right)^i \mathbf{v}_{k_i}$$

is the usual binary expansion, valid for $\mathbf{x} \in [-2, 2]^N$. Hence the binary expansion is a particular case of an α-expansion. This is also true for the expansion of real numbers in any integer basis $q > 2$.

There is an interesting relation between α-expansions and the theory of iterated function systems (IFS) [1,2,11]. It occurs that the set of points $\mathbf{x} \in \mathbf{R}^N$ representable in the form (1) is the attractor of an IFS. This attractor is a fractal set when α is small, but contains open sets when α is large. Therefore, it is important to find the critical value of α when this attractor stops to be fractal and starts to contain open sets. In general, finding this critical value is difficult, but we can make some estimates of it. These are dealt with in Section 2. We also propose an algorithm, based on the IFS theory, to obtain the sequence $(\mathbf{v}_{k_0}, \mathbf{v}_{k_1}, \ldots)$ that represents a point $\mathbf{x} \in \mathbf{R}^N$. The drawback of the algorithm is that it requires a way of determining if a point is in the convex hull of some given points in $\mathbf{R}^N$, which is computationally hard, if N is large.

In Section 3, we propose another algorithm for obtaining α-expansions, which we call the *nearest point algorithm.* This algorithm is a greedy one and may be seen as a quantized version of the matching pursuits algorithm [6,8,9]. One problem of this algorithm is that the set of points for which it converges is in general smaller than the set of points where (1) holds. In fact, for some codebooks, the nearest point algorithm does not converge in an open set for any value of α, which means that it is not useful for such codebooks. However, if the codebook is "symmetric," in a sense defined in Section 3, then its convergence set coincides with the set where (1) holds.

The α-expansions were originally used for vector quantization of wavelet transform coefficients in the context of image coding [4,13]. In [5], we show that they can be useful for coding frame coefficients. In this article, we further develop this latter application.

The usual scheme for coding the coefficients of a frame expansion is to calculate the coefficients first, quantize them in a second step and reconstruct the vector in a third step. We shall call any such scheme a decompose-quantize procedure. In Section 4, we compare the rate-distortion characteristics of a general decompose-quantize procedure and the α-expansions algorithm. In this comparison, we don't consider entropy coding. We show that the α-expansion is, for high rate coding, better than any decompose-quantize procedure if

$$\frac{\log_2(2p)}{\log_2(\frac{1}{\alpha})} < p, \tag{4}$$

where p is the number of elements in the frame. We shall also give examples where inequality (4) is strict.

2. Theory of α expansions

Consider a codebook $\mathcal{C} = \{\mathbf{v}_1, \ldots, \mathbf{v}_q\}$ of vectors in $\mathbf{R}^N$ and let α be a number in the interval (0,1). In this section, we develop the theory of α-expansions over $\mathcal{C}$. We shall denote by $\Lambda = \Lambda(\alpha, \mathcal{C})$ the set of points of $\mathbf{R}^N$ that can be represented in the form (1) and by $P = P(\alpha, \mathcal{C})$ the convex hull of the vectors $\{\mathbf{w}_1, \ldots, \mathbf{w}_q\}$, where for $1 \leqslant k \leqslant q$,

$$\mathbf{w}_k = \frac{\mathbf{v}_k}{1-\alpha}. \tag{5}$$

2.1. Iterated function system

We say that a mapping $f : P \to P$ is contractive if there exists $0 < \gamma < 1$ such that

$$\left\| f(\mathbf{x}) - f(\mathbf{y}) \right\| \leqslant \gamma \|\mathbf{x} - \mathbf{y}\|.$$

An iterated function system (IFS) on P is a finite collection of contractive maps of P into itself [1,2].

An α-expansion is closely related to an iterated function system. The mappings of this IFS are very simple and have a geometrical interpretation. They are α-similarities with center $\mathbf{w}_k$. We define the mappings f_k, $1 \leqslant k \leqslant q$, by the formula

$$f_k(\mathbf{x}) = \alpha \mathbf{x} + \mathbf{v}_k.$$

Proposition 2.1. *The mappings f_k, $1 \leqslant k \leqslant q$, define an iterated function system on $P = P(\alpha, \mathcal{C})$.*

Proof. Observe first that

$$f_k(\mathbf{x}) = \alpha(\mathbf{x} - \mathbf{w}_k) + \mathbf{w}_k,$$

where $\mathbf{w}_k$ is defined by (5). Therefore f_k is an α similarity of center $\mathbf{w}_k$. This shows that $f_k(P) \subset P$. Since each f_k is contractive, we conclude that the mappings f_k, $1 \leqslant k \leqslant q$, define in fact an IFS on P. □

Every IFS on a compact set P has an attractor defined by $\bigcap_{n=0}^{\infty} F^n(P)$, where F is a set function defined by $F(A) = f_1(A) \cup \cdots \cup f_q(A)$.

Proposition 2.2. *The attractor of the IFS is exactly $\Lambda = \Lambda(\alpha, \mathcal{C})$.*

Proof. Observe first that any $\mathbf{x} \in \Lambda$ is a linear combination of the vectors $\{\mathbf{v}_k\}_{k=1,\ldots,q}$ with coefficients whose sum is exactly $1/(1-\alpha)$, and therefore must belong to P. Hence $\Lambda \subset P$.

If $\mathbf{x} \in \Lambda$, then we can write, for any $n \geqslant 0$,

$$\mathbf{x} = \mathbf{v}_{k_0} + \alpha \mathbf{v}_{k_1} + \cdots + \alpha^{n-1} \mathbf{v}_{k_{n-1}} + \alpha^n \mathbf{y} \tag{6}$$

for some $\mathbf{y} \in P$. This is equivalent to

$$\mathbf{x} = f_{k_0} \circ f_{k_1} \circ \cdots \circ f_{k_{n-1}}(\mathbf{y}), \tag{7}$$

which implies that $\mathbf{x} \in \bigcap_{j=0}^{n-1} F^j(P)$. Since this holds for any $n \geqslant 0$, we conclude that $\mathbf{x}$ must be in the attractor of the IFS. On the other hand, if $\mathbf{x} \in \bigcap_{n=0}^{\infty} F^n(P)$, we can write (7) for any $n \geqslant 0$, and some $\mathbf{y} \in P$. But formula (7) is equivalent to formula (6), and hence (6) holds for every $n \geqslant 0$. We conclude that $\mathbf{x} \in \Lambda$. □

In general, the attractor of an IFS is a fractal set (see Fig. 1). However, we are interested in the case where it contains an open set. An important case is when the attractor is the whole convex hull P. This case occurs if and only if $F(P) = P$.

Example 2.3. Consider the case where the codebook is $\mathcal{B}_N$ defined by (3) and $\alpha = \frac{1}{2}$. We have that $P(\frac{1}{2}, \mathcal{B}_N) = [-2, 2]^N$ and one can see that

$$f_{(\varepsilon_1,\ldots,\varepsilon_N)}(P) = I_{\varepsilon_1} \times \cdots \times I_{\varepsilon_N},$$

where $I_1 = [0, 2]$, $I_{-1} = [-2, 0]$, and $\times$ denotes the Cartesian product. It is now easy to conclude that $F(P) = P$, which implies that $\Lambda = P$.

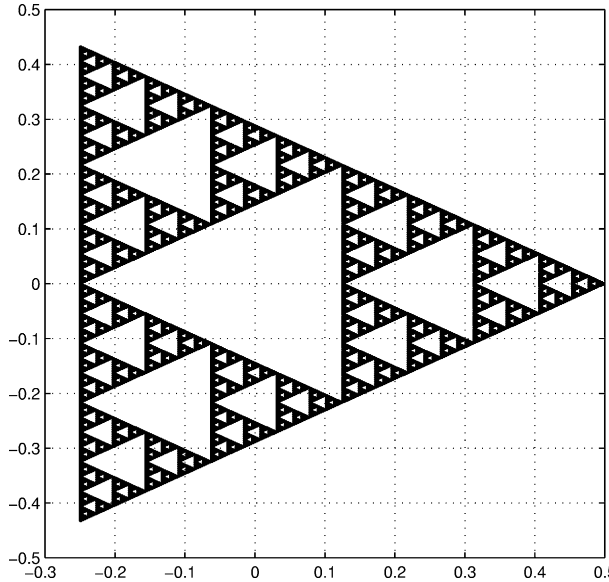

Fig. 1. The set Λ for $\mathcal{C} = \{(\frac{1}{4}, 0), (-\frac{1}{8}, \frac{\sqrt{3}}{8}), (-\frac{1}{8}, -\frac{\sqrt{3}}{8})\}$ and $\alpha = \frac{1}{2}$.

It can occur that Λ contains an open set without being equal to P, as the next example shows us.

Example 2.4. Let $\mathcal{C} = \mathcal{B}_3 \cup \{(2, 0, 0)\}$. If we consider $\alpha = \frac{1}{2}$, then $[-2, 2]^3 \subset \Lambda$, since by Example 1 every $\mathbf{x} \in [-2, 2]^3$ can be represented as an $(\frac{1}{2}, \mathcal{B}_3)$ expansion. But Λ does not contain entirely any triangular face of P, since by Proposition 2.10 such a face will only be entirely covered with $\alpha = \frac{2}{3}$.

In all the examples we have considered, if Λ contains an open set, it contains also the convex hull of some subset of $\{\mathbf{w}_1, \dots, \mathbf{w}_q\}$. We believe that this is a general fact and so we formulate it as a conjecture.

Conjecture 2.5. *If Λ contains an open set of $\mathbf{R}^N$, then it contains a convex hull of some subset of $\{\mathbf{w}_1, \dots, \mathbf{w}_q\}$.*

2.2. The choice of the parameter α

We shall now consider the problem of finding a good value of α for a given codebook $\mathcal{C}$. We shall use the notation $\Lambda_\alpha = \Lambda(\alpha, \mathcal{C})$, $P_\alpha = P(\alpha, \mathcal{C})$, $f_{k,\alpha} = f_k(\alpha, \mathcal{C})$, and $F_\alpha = F(\alpha, \mathcal{C})$. Our interest is to find α as small as possible such that $\Lambda_\alpha = P_\alpha$.

Proposition 2.6. *If $\Lambda_{\alpha_1} = P_{\alpha_1}$, then $\Lambda_{\alpha_2} = P_{\alpha_2}$ for any $\alpha_2 \geqslant \alpha_1$.*

Proof. In order to compare subsets of P_α for different values of α, we first perform a $(1-\alpha)$-similarity on them. Observe that $(1-\alpha)P_\alpha$ is exactly the convex hull of $\{\mathbf{v}_1, \dots, \mathbf{v}_q\}$, that we shall denote P_0.

We have that, for any $1 \leqslant k \leqslant q$, $f_{k,\alpha}(P_\alpha)$ is obtained from P_α by an α similarity of center $\mathbf{w}_k$. Therefore $(1-\alpha) f_{k,\alpha}(P_\alpha)$ is obtained from P_0 by an α similarity of center $\mathbf{v}_k$. Hence, for $\alpha_1 \leqslant \alpha_2$,

$$(1-\alpha_1) f_{k,\alpha_1}(P_{\alpha_1}) \subset (1-\alpha_2) f_{k,\alpha_2}(P_{\alpha_2}),$$

and so $(1-\alpha_1)F_{\alpha_1}(P_{\alpha_1}) \subset (1-\alpha_2)F_{\alpha_2}(P_{\alpha_2})$.

Since, by hypothesis, $F_{\alpha_1}(P_{\alpha_1}) = P_{\alpha_1}$, we conclude that $P_0 \subset (1-\alpha_2)F_{\alpha_2}(P_{\alpha_2})$. Hence $F_{\alpha_2}(P_{\alpha_2}) = P_{\alpha_2}$ and therefore $\Lambda_{\alpha_2} = P_{\alpha_2}$. $\square$

The proposition above implies the existence of a critical value of α, that we shall denote α_c, such that $\Lambda_\alpha = P_\alpha$ if and only if $\alpha \geqslant \alpha_c$. As an example, observe that the critical value of α for the codebook $\mathcal{B}_N$ is $\alpha_c = \frac{1}{2}$.

The next proposition gives us a lower bound for α_c.

Proposition 2.7. *If $\Lambda = \Lambda(\alpha, \mathcal{C})$ contains an open set of $P = P(\alpha, \mathcal{C})$, then $q\alpha^N \geqslant 1$. As a consequence*

$$\alpha_c \geqslant \left(\frac{1}{q}\right)^{1/N}.$$

Proof. We shall denote the N-dimensional volume of a set Λ by $m(A)$. Let B be an open set contained in Λ. We have that, for any $n \geqslant 0$, $F^n(P)$ is a union of at most q^n sets of N-dimensional volumes of at most $(\alpha^N)^n m(P)$. Hence

$$\left(q\alpha^N\right)^n m(P) \geqslant m\left(F^n(P)\right) \geqslant m(B).$$

Since this must hold for any $n \geqslant 0$, we conclude that $q\alpha^N \geqslant 1$. □

We now proceed to find an upper bound for α_c. We shall first consider the particular case when the codebook has exactly $N+1$ vectors not contained in a hyperplane of $\mathbf{R}^N$.

Suppose that the vectors of the codebook $\mathcal{C} = \{\mathbf{v}_1, \dots, \mathbf{v}_{N+1}\}$ are not in a hyperplane of $\mathbf{R}^N$. Then the convex hull P of $\{\mathbf{w}_1, \dots, \mathbf{w}_{n+1}\}$, $\mathbf{w}_k = \mathbf{v}_k/(1-\alpha)$ is a $(N+1)$-polytope in $\mathbf{R}^N$ and every point $\mathbf{x} \in P$ can be written in a unique way as

$$\mathbf{x} = \sum_{k=1}^{N+1} t_k \mathbf{w}_k,$$

where $0 \leqslant t_k \leqslant 1$ and $\sum_{k=1}^{N+1} t_k = 1$. The vector $(t_1, \dots, t_{N+1})$ is called the vector of barycentric coordinates of $\mathbf{x}$. The barycenter $\mathbf{x}$ of P is the point whose vector of barycentric coordinates is $(1/(N+1), \dots, 1/(N+1))$.

Lemma 2.8. *In barycentric coordinates, the contractive map f_k is defined by*

$$f_k(t_1, \dots, t_k, \dots, t_{N+1}) = (\alpha t_1, \dots, \alpha t_k + 1 - \alpha, \dots, \alpha t_{N+1}),$$

for any $1 \leqslant k \leqslant N+1$.

Proof. If $\mathbf{x} = (t_1, \dots, t_k, \dots, t_{N+1})$, we have that

$$\begin{aligned} f_k(\mathbf{x}) &= \alpha \sum_{j=1}^{N+1} t_j \mathbf{w}_j + (1-\alpha)\mathbf{w}_k \\ &= \sum_{j \neq k} \alpha t_j \mathbf{w}_j + (\alpha t_k + 1 - \alpha)\mathbf{w}_k, \end{aligned}$$

thus proving the lemma. □

Proposition 2.9. *Assuming that $q = N+1$ and that $\{\mathbf{v}_1, \dots, \mathbf{v}_{N+1}\}$ are not contained in a hyperplane of $\mathbf{R}^N$, we have that $\Lambda_\alpha = P_\alpha$ if and only if*

$$\alpha \geqslant \frac{N}{N+1}.$$

Proof. Observe first that if $\alpha < N/(N+1)$, then

$$\alpha t_k + 1 - \alpha \geqslant 1 - \alpha > \frac{1}{N+1},$$

which implies that $\mathbf{b} \notin f_k(P_\alpha)$. Since this holds for any $1 \leqslant k \leqslant N+1$, we conclude that $\mathbf{b} \notin F_\alpha(P_\alpha)$. Hence $\Lambda_\alpha \neq P_\alpha$.

On the other hand, suppose that $\alpha \geqslant N/(N+1)$. Any $\mathbf{y} = (s_1, \ldots, s_k, \ldots, s_{N+1}) \in P_\alpha$ must have a coordinate, say s_k, larger than or equal to $1/(N+1)$. And all others must then be smaller than $N/(N+1)$. Take

$$\mathbf{x} = \left(\frac{s_1}{\alpha}, \ldots, \frac{s_k - 1 + \alpha}{\alpha}, \ldots, \frac{s_{N+1}}{\alpha}\right).$$

Observe that $\alpha \geqslant N/(N+1)$ implies that all coordinates of $\mathbf{x}$ are in the interval $[0, 1]$, and hence $\mathbf{x} \in P_\alpha$. Since $f_k(\mathbf{x}) = \mathbf{y}$, this implies that $\mathbf{y} \in f_k(P_\alpha)$. We conclude that

$$\bigcup_{j=1}^{N+1} f_j(P_\alpha) = P_\alpha$$

and hence $\Lambda_\alpha = P_\alpha$. $\square$

Corollary 2.10. *For any codebook in $\mathbf{R}^N$, if $\alpha \geqslant N/(N+1)$ then $\Lambda_\alpha = P_\alpha$.*

Proof. This corollary is a consequence of Carathéodory's fundamental theorem [7], which states that each point in the convex hull os a set S in $\mathbf{R}^N$ is the convex combination of $N+1$ or fewer points of S. By the proposition above, if we take $\alpha \geqslant N/(N+1)$, $F_\alpha(P_\alpha)$ contains all the convex hulls of $N+1$ points of $\mathcal{C}$. Therefore $F_\alpha(P_\alpha)$ must contain P_α itself. This implies that $\Lambda_\alpha = P_\alpha$. $\square$

This corollary gives us an upper bound for α_c. In fact, we have that

$$\alpha_c \leqslant \frac{N}{N+1}.$$

2.3. *The inverse mapping and an algorithm for α-expansions*

In this subsection, we shall describe an algorithm for obtaining an α-expansion of a point $\mathbf{x} \in \Lambda$, assuming that α was chosen with the property that $\Lambda = P$. With this assumption, every $\mathbf{x} \in P$ can be represented in the form (1).

The problem here is that the sequence $(k_0, k_1, \ldots)$ of indexes of vectors in the codebook associated to a given $\mathbf{x} \in P$ is not unique in general. This ambiguity occurs because a $\mathbf{x} \in P$ can belong to $f_k(P)$ for different values of k.

Denote by $\mathcal{I}(\mathbf{x})$ the set of indexes k such that $\mathbf{x} \in f_k(P)$. In order to eliminate the ambiguity, we must select for each $\mathbf{x}$ a unique $K(\mathbf{x}) \in \mathcal{I}(\mathbf{x})$. So we define $K(\mathbf{x})$ to be the index $k \in \mathcal{I}(\mathbf{x})$ satisfying the property

$$\mathrm{d}(\mathbf{x}, \mathbf{v}_k) \leqslant \mathrm{d}(\mathbf{x}, \mathbf{v}_j),$$

for any $j \in I(\mathbf{x})$. If there are more than one such k, choose $K(\mathbf{x})$ to be the smallest one.

The function K determines a mapping $g : P \to P$ defined by

$$g(\mathbf{x}) = \frac{\mathbf{x} - \mathbf{v}_{K(\mathbf{x})}}{\alpha}.$$

It is clear that $f_{K(\mathbf{x})} \circ g(\mathbf{x}) = \mathbf{x}$ and if $K(f_k(\mathbf{x})) = k$, $g \circ f_k(\mathbf{x}) = \mathbf{x}$. Because of these properties, we shall call g the inverse mapping of the IFS.

Example 2.11 (Continuation of Example 2.3). In the case $\mathcal{C} = \mathcal{B}_N$ and $\alpha = \frac{1}{2}$, the index set $\mathcal{I}(\mathbf{x})$ is unitary for almost all $\mathbf{x} \in P$. And the inverse mapping g restricted to $I_{\varepsilon_1} \times \cdots \times I_{\varepsilon_N}$ is a two-similarity of center $\mathbf{w} = 2(\varepsilon_1, \ldots, \varepsilon_N)$.

We shall denote by g^n the nth iteration of the mapping g. In other terms, g^n is defined inductively by the formula $g^n(\mathbf{x}) = g(g^{n-1}(\mathbf{x}))$.

Proposition 2.12. *For any $\mathbf{x} \in P$, choose $k_n = K(g^n(\mathbf{x}))$. Then the residual vector $\mathbf{r}_n(\mathbf{x})$ is given by the formula*

$$\mathbf{r}_n(\mathbf{x}) = \alpha^n g^n(\mathbf{x}). \tag{8}$$

Proof. We shall prove the proposition by induction. For $n = 1$, Eq. (8) is just the definition of the mapping g. Assuming that Eq. (8) holds for $n = t$, we shall prove that it also holds for $n = t + 1$. But

$$\alpha^{t+1} g^{t+1}(\mathbf{x}) = \alpha^t \left(g^t(\mathbf{x}) - \mathbf{v}_{k_t}\right) = \mathbf{r}_t(\mathbf{x}) - \alpha^t \mathbf{v}_{k_t} = \mathbf{r}_{t+1}(\mathbf{x}),$$

thus proving the proposition. □

We can now describe an algorithm for obtaining the sequence $(k_0, k_1, \ldots)$ of indexes that represents the α-expansion of a vector $\mathbf{x} \in P(\alpha, \mathcal{C})$. It is a recursive algorithm, where for each $n \geqslant 0$ we compute

$$k_n = K\left(g^n(\mathbf{x})\right).$$

By the above proposition $\|\mathbf{r}_n(\mathbf{x})\|$ converges to 0 and hence the α-expansion of $\mathbf{x}$ converges.

The computation of $K(\mathbf{x})$ requires a practical way to determine if $\mathbf{x}$ belongs to the convex hull of a given set of vectors. If the dimension N of the space is large, solving this problem is computationally intractable. Hence this algorithm is not practical if N is large, and in the next section we shall describe a feasible algorithm for obtaining the α-expansion of a vector $\mathbf{x} \in P$.

2.4. Rate-distortion characteristics of the α-expansions

It is sometimes important to have an estimate of the error in the approximation by an α-expansion with a given number of parcels. In what follows, we compute a bound on the error of the approximation of a vector $\mathbf{x} \in \mathbf{R}^N$, given the rate spent in representing the sequence of integers $(k_0, k_1, \ldots, k_n)$.

Assume that $\Lambda(\alpha, \mathcal{C}) = P(\alpha, \mathcal{C})$. If we approximate $\mathbf{x} \in P(\alpha, \mathcal{C})$ by its n-term expansion $(\mathbf{v}_{i_0}, \mathbf{v}_{i_1}, \ldots, \mathbf{v}_{i_{n-1}})$, the error of the approximation is given by $\mathbf{r}_n(\mathbf{x})$ (see Eq. (2)). Hence the mean squared distortion is given by

$$D = \int_P \left\|\mathbf{r}_n(\mathbf{x})\right\|^2 \mathrm{d}m(\mathbf{x}), \tag{9}$$

where m denotes the usual N-dimensional volume [10].

Example 2.13 (Continuation of Example 2.3). In the case $\mathcal{C} = \mathcal{B}_N$ and $\alpha = \frac{1}{2}$, the residual $\mathbf{r}_n(\mathbf{x})$ is given by $\mathbf{r}_n(\mathbf{x}) = (\frac{1}{2})^n g^n(\mathbf{x})$ (see Proposition 2.12). Since m is a g-invariant measure, we conclude that

$$D = \left(\frac{1}{2}\right)^{2n} \int_P \|\mathbf{x}\|^2 \mathrm{d}m(\mathbf{x}).$$

It is easy to calculate the last integral and obtain

$$D = \frac{N 2^N}{3} \left(\frac{1}{2}\right)^{2n}.$$

In the general case, the mean squared distortion is not easy to calculate, but we can compute an upper bound for it. Since $\mathbf{r}_n(\mathbf{x}) \in P(\alpha, \mathcal{C})$, we have that $\|\mathbf{r}_n(\mathbf{x})\| \leqslant B/(1-\alpha)\alpha^n$, where B is the maximum norm of a vector in $\mathcal{C}$. So we conclude from (9) that

$$D \leqslant \frac{B^2 m(P(\alpha, \mathcal{C}))}{(1-\alpha)^2} \alpha^{2n}.$$

The total number of bits needed to represent $(k_0, k_1, \ldots, k_{n-1})$ is at most $n \log_2 q$. Therefore the rate-distortion function $R(D)$ must satisfy the inequality

$$R(D) \leqslant \frac{\log_2(q)}{2 \log_2(\frac{1}{\alpha})} \log_2\left(\frac{C}{D}\right), \tag{10}$$

where $C = B^2 m(P(\alpha, \mathcal{C}))/(1-\alpha)^2$ is a constant.

3. Nearest point algorithm

In Section 2.3 we have described an algorithm to obtain the α-expansion of a vector, but it is computationally very expensive. In this section, we shall see a more practical algorithm for this task, referred to as the nearest point algorithm.

3.1. Description and convergence of the algorithm

For any $\mathbf{x} \in \mathbf{R}^N$, denote by $K_0(\mathbf{x})$ the index of the codevector $\mathbf{v}_k \in \mathcal{C}$ nearest to $\mathbf{x}$. If there are more than one choice for $\mathbf{v}_k$, choose the one with the smallest index. More formally, denoting by V_k, the intersection of the Voronoi cell of $\mathbf{v}_k$ with P,

$$V_k = \left\{\mathbf{x} \in P \;\middle|\; \mathrm{d}(\mathbf{x}, \mathbf{v}_k) \leqslant \mathrm{d}(\mathbf{x}, \mathbf{v}_j),\ \text{for any } 1 \leqslant j \leqslant q\right\},$$

we define $K_0(\mathbf{x})$ to be the smallest $k \in \{1, \dots, q\}$ such that $\mathbf{x} \in V_k$.

In order to describe the nearest point algorithm, we define the mapping $g_0 : \mathbf{R}^N \to \mathbf{R}^N$ by

$$g_0(\mathbf{x}) = \frac{\mathbf{x} - \mathbf{v}_{K_0(\mathbf{x})}}{\alpha}.$$

This mapping has some properties similar to the inverse mappings of Section 2.3. In fact, we have that $f_{K_0(\mathbf{x})} \circ g(\mathbf{y}) = \mathbf{y}$ and if $K_0(f_k(\mathbf{x})) = k$, $g \circ f_k(\mathbf{x}) = \mathbf{x}$.

To find the sequence $(k_0, k_1, \dots)$ of indexes that represents the α-expansion of a vector $\mathbf{x} \in \mathbf{R}^N$, the nearest point algorithm recursively iterates g_0 obtaining the sequence $k_i = K_0(g_0^i(\mathbf{x}))$, $i \geqslant 0$. We say that the algorithm converges for the vector $\mathbf{x}$ if the norm of residual vector $\mathbf{r}_n(\mathbf{x})$ goes to zero, as n goes to infinite.

Denote by $\Lambda_0 = \Lambda_{0,\alpha} \subset P$ the set of points $\mathbf{x} \in P$ that remains in P under any number of interactions of g_0, i.e.,

$$\Lambda_0 = \bigcap_{n=0}^{\infty} g_0^{-n}(P).$$

It is a nice fact that this set is exactly the convergence set of the nearest point algorithm.

Proposition 3.1. *Take any $\mathbf{x} \in \mathbf{R}^N$. The nearest point algorithm converges for $\mathbf{x}$ if and only if $\mathbf{x} \in \Lambda_0$.*

Proof. By using a similar reasoning as the one used in Proposition 2.12, one can show that

$$\mathbf{r}_n(\mathbf{x}) = \alpha^n g_0^n(\mathbf{x}),$$

for any $\mathbf{x} \in \mathbf{R}^N$ and $n \geqslant 0$. If $\mathbf{x} \in \Lambda_0$, then $g_0^n(\mathbf{x}) \in P$, for every $n \geqslant 0$, and hence we conclude that $\|\mathbf{r}_n(\mathbf{x})\|$ converges to zero. This shows that the nearest point algorithm converges for $\mathbf{x}$.

On the other hand, if $\|\mathbf{r}_n(\mathbf{x})\|$ converges to zero, then

$$g_0^n(\mathbf{x}) = \frac{1}{\alpha^n}\left(\sum_{i=n}^{\infty} \alpha^i \mathbf{v}_{k_i}\right) = \sum_{j=0}^{\infty} \alpha^j \mathbf{v}_{k_{j+n}}$$

belongs to P, for any $n \geqslant 0$. Therefore $\mathbf{x} \in \Lambda_0$. $\square$

Proposition 3.1 above implies that $\Lambda_0 \subset \Lambda$. But it is possible that $\Lambda_0 \neq \Lambda$, as illustrated in the next example.

Example 3.2. Let $\mathcal{C} = \{\mathbf{v}_1 = (1, 0),\ \mathbf{v}_2 = (0, 1),\ \mathbf{v}_3 = (-1, 0)\}$. For any $0 < \alpha < 1$, there exists $\delta > 0$ such that the segment $(0, \delta)$ is contained in $P \setminus f_2(P)$. Since this segment is in the Voronoi cell V_2, we conclude that $g_0(0, \delta)$ is not in P. Therefore the segment is not in Λ_0. We conclude that $\Lambda_0 \subsetneqq P$, for any $0 < \alpha < 1$. On the other hand, if $\alpha \geqslant \frac{2}{3}$, Proposition 2.9 says that $\Lambda = P$.

3.2. Choice of α for the convergence of the nearest point algorithm

In this section, we shall assume that α was chosen with the property that $\Lambda = P$. As the example above shows us, this does not guarantees that $\Lambda_0 = P$.

The next proposition gives us a necessary and sufficient condition in order to have $\Lambda_0 = P$.

Proposition 3.3. *$\Lambda_0 = P$ if and only if $V_k \subset f_k(P)$, for every $1 \leqslant k \leqslant q$.*

Proof. If $V_k \subset f_k(P)$, for every $1 \leqslant k \leqslant q$, then g_0 is exactly the inverse mapping defined in Section 2.

On the other hand, if there exists $k \in \{1, \dots, q\}$ and a point $\mathbf{y} \in V_k$, with $\mathbf{y} \notin f_k(P)$, then $g_0(\mathbf{y}) \notin P$ and therefore $\mathbf{y} \notin \Lambda_0$. We conclude that if $\Lambda_0 = P$ then $V_k \subset f_k(P)$, for every $1 \leqslant k \leqslant q$. □

For a given codebook $\mathcal{C}$, α_c is the smallest value of α such that $\Lambda = P$. If, taking $\alpha = \alpha_c$, we have the property $V_k \subset f_k(P)$, for every $1 \leqslant k \leqslant q$, then also $\Lambda_0 = P$. This last property can be considered a kind of symmetry of the codebook $\mathcal{C}$. In order to use the nearest point algorithm, these "symmetric" codebooks are preferable.

In the case that the codebook is not "symmetric," we must choose a larger value of α in order to the nearest point algorithm to converge for any $\mathbf{x} \in P$. We have the following proposition.

Proposition 3.4. *If $\Lambda_{0,\alpha_1} = P_{\alpha_1}$ and $\alpha_1 \geqslant \alpha_1$, then $\Lambda_{0,\alpha_2} = P_{\alpha_2}$.*

Proof. If $\Lambda_{0,\alpha_1} = P_{\alpha_1}$, then, by Proposition 3.3, $V_{k,\alpha_1} \subset f_{k,\alpha_1}(P_{\alpha_1})$, for every $1 \leqslant k \leqslant q$. But $(1-\alpha_1)V_{k,\alpha_1} = (1-\alpha_2)V_{k,\alpha_2}$ and $(1-\alpha_1)f_{k,\alpha_1}(P_{\alpha_1}) \subset (1-\alpha_2)f_{k,\alpha_2}(P_{\alpha_2})$. Therefore $V_{k,\alpha_2} \subset f_{k,\alpha_2}(P_{\alpha_2})$, for every $1 \leqslant k \leqslant q$, and hence $\Lambda_{0,\alpha_2} = P_{\alpha_2}$. □

The above proposition shows that there exists a critical value of α, denoted by $\alpha_{0,c}$, for which $\Lambda_{0,\alpha} = P_\alpha$ if and only if $\alpha \geqslant \alpha_{0,c}$. And it is clear that $\alpha_{0,c} \geqslant \alpha_c$. On the other hand, in Example 3.2 we have seen that $\Lambda_{0,\alpha} \subsetneqq P_\alpha$, for any $0 < \alpha < 1$. Hence we cannot guarantee that $\alpha_{0,c} < 1$. If $\mathcal{C}$ is a codebook such that $\alpha_{0,c} = 1$, then it cannot be used for α-expansions, at least using the nearest point algorithm.

4. Application to frame expansions

In this section, we shall look at α-expansions as frame expansions whose coefficients are already quantized, and thus are ready for coding. We shall argue that in many cases using α-expansions is better, in a rate-distortion sense, for high bit rate coding than any decompose-quantize procedure.

4.1. Frames and α-expansions

Let $\mathcal{F} = \{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_p\}$ be a collection of vectors generating $\mathbf{R}^N$. This means that every $\mathbf{x} \in \mathbf{R}^N$ can be expressed as

$$\mathbf{x} = \sum_{i=1}^{p} a_i \mathbf{e}_i, \tag{11}$$

for some coefficients $a_i, \dots, a_p$. The vectors $\{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_p\}$ may or not be linearly independent. In the case that they are linearly dependent, the set $\mathcal{F}$ is called a frame or an overcomplete basis. In this section, we shall call $\mathcal{F}$ a frame even if the vectors $\{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_p\}$ are linearly independent. More on frames can be found in [12].

Given a frame $\mathcal{F} = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_p\}$, consider the codebook $\mathcal{C} = \{\mathbf{v}_1, \ldots, \mathbf{v}_q\}$, where $q = 2p$ and, for $1 \leqslant i \leqslant p$, $\mathbf{v}_i = \mathbf{e}_i$, and $\mathbf{v}_{i+p} = -\mathbf{e}_i$. Then the α-expansion over $\mathcal{C}$ of a vector $\mathbf{x}$ given by

$$\mathbf{x} = \sum_{i=0}^{\infty} \alpha^i \mathbf{v}_{k_i} \tag{12}$$

can be seen as a decomposition of $\mathbf{x}$ in the frame $\mathcal{F}$.

We observe that in Eq. (11), the sum is finite, while in Eq. (12) it is infinite. At a first sight this would give a disadvantage for the α-expansions from a coding point of view. However, in frame decompositions as (11), we represent $\mathbf{x} \in \mathbf{R}^N$ by the p-tuple $(a_1, a_2, \ldots, a_p)$, where the a_is are real numbers and hence they must be quantized, introducing an error. In the case of (12), we represent $\mathbf{x} \in \mathbf{R}^N$ by the sequence $(k_0, k_1, \ldots)$ of integers, that has to be truncated, also introducing an error. Therefore there is no a prior disadvantage of α-expansions in this respect.

4.2. Rate-distortion characteristics of the decompose-quantize procedure

A general decompose-quantize procedure works as follows:

1. In the first step, the frame coefficients $(a_1, a_2, \ldots, a_p)$ are obtained from the vector $\mathbf{x} \in P \subset \mathbf{R}^N$ (see [12]). The vector $(a_1, a_2, \ldots, a_p)$ is contained in a compact subset S of an N-dimensional subspace $L = L(\mathcal{F})$ of $\mathbf{R}^p$.
2. In the second step, the vector $(a_1, a_2, \ldots, a_p)$ is uniformly quantized to $(a_1^n, a_2^n, \ldots, a_p^n)$, where each a_i^n is represented by n bits. The quantization cells are the hypercubes
$$\left[\frac{i_1}{2^n}, \frac{i_1+1}{2^n}\right] \times \cdots \times \left[\frac{i_p}{2^n}, \frac{i_p+1}{2^n}\right], \quad i_j \in \mathbf{Z}.$$
We shall denote by $\mathcal{H}_n$ the partition of $\mathbf{R}^p$ in the above hypercubes.
3. In the third and last step, a frame reconstruction scheme recovers a vector $\mathbf{y}$ near to $\mathbf{x}$ from the quantized coefficients $(a_1^n, a_2^n, \ldots, a_p^n)$. We shall denote by f the initial probability distribution function of the points $\mathbf{x} \in P$ and *assume that f is absolutely continuous*. The mean squared error D_n of the scheme is the mean of $\|\mathbf{y} - \mathbf{x}\|$ with respect to f. It was proved in [16] and [17] that there exists a coefficient $c(N, f)$ such that, for n large enough,
$$D_n \geqslant \frac{c(N, f)}{F_n^{2/N}}, \tag{13}$$
where F_n denotes the number of quantization cells in the polytope P.

Lemma 4.1. *There exists a constant $b = b(S, L)$ such that*

$$\#(\mathcal{H}_n \cap S) \leqslant b 2^{nN}.$$

Proof. Without loss of generality, we can assume that the orthogonal projection π of the subspace L in

$$L_N = \left\{(x_1, x_2, \ldots, x_p) \;\middle|\; x_{N+1} = \cdots = x_p = 0\right\}$$

is injective. In other words, the subspace L can be defined by the $(p - N)$ equations

$$x_{N+i} = \sum_{j=1}^{N} l_{ij} x_j, \quad 1 \leqslant i \leqslant p.$$

The partition $\mathcal{H}_n$ of $\mathbf{R}^p$ determines a partition $\pi(\mathcal{H}_n)$ of L_N. And it is not difficult to see that, for any cell $B \in \pi(\mathcal{H}_n)$, the number of cells of $\mathcal{H}_n$ whose projection is B and intersecting L is at most $l^* = \max\{|l_{ij}|,\ 1 \leqslant i \leqslant p,\ 1 \leqslant j \leqslant N\} + 1$.

Also, if n is large, the number of cells of $\pi(\mathcal{H}_n)\cap\pi(S)$ is less than $(m(\pi(S))+1)2^{nN}$, where m denotes the N-dimensional Lebesgue measure. Hence the number of cells of $\mathcal{H}_n$ intersecting L is at most $l^*(m(\pi(S))+1)2^{nN}$. So we conclude that

$$\#(\mathcal{H}_n\cap S)\leqslant b2^{nN},$$

where $b(S,L)=l^*(m(\pi(S))+1)$. □

Corollary 4.2. *Assume that the initial probability distribution f in the polytope P is absolutely continuous. Then there is a constant $C=C(\mathcal{F},\alpha,f)$ such that*

$$D_n\geqslant C\left(\frac{1}{2}\right)^{2n}. \tag{14}$$

Proof. Observe first that $F_n\leqslant\#(\mathcal{H}_n\cap S)$. Then, substituting the result of Lemma 4.1 in (13) and making

$$C=\frac{c(N,f)}{b^{2/N}},$$

we obtain the desired result. □

Now we can compute a lower bound of the rate-distortion characteristics of the general decompose-quantize procedure. *We shall not consider entropy coding for the vector* $(a_1^n,a_2^n,\dots,a_p^n)$. So the total number of bits used to represent $(a_1^n,a_2^n,\dots,a_p^n)$ is $R=np$. Therefore, using (14), we can write

$$R(D)\geqslant\frac{p}{2}\log_2\left(\frac{C}{D}\right), \tag{15}$$

which is the bound we were looking for.

4.3. Comparison between α-expansions and the decompose-quantize procedure

In this section we make a theoretical comparison between α-expansions and the decompose-quantize procedure. In this comparison, both schemes are considered without entropy coding.

Proposition 4.3. *Assume that the initial probability distribution in the polytope P is absolutely continuous. If inequality* (11) *holds, i.e., if*

$$\frac{\log_2(2p)}{\log_2(\frac{1}{\alpha})}<p,$$

then the α-expansion is better, in a rate-distortion sense, than the decompose-quantize procedure for high bit rate coding.

Proof. Let us compare (15) and (10), for a fixed distortion D. We have that

$$\frac{R_2(D)}{R_1(D)}\leqslant\left[\frac{\log_2(2p)}{p\log_2(\frac{1}{\alpha})}\right]\left[\frac{\log_2(\frac{C_2}{D})}{\log_2(\frac{C_1}{D})}\right],$$

where the indexes 1 and 2 in R and C are associated to the decompose-quantize procedure and the α-expansions, respectively. For high rate, $D\ll C_1$ and $D\ll C_2$, and hence the factor inside the second brackets is close to 1. Since inequality (11) says that the factor inside the first brackets is smaller than 1, we conclude that

$$\frac{R_2(D)}{R_1(D)}<1,$$

thus proving the proposition. □

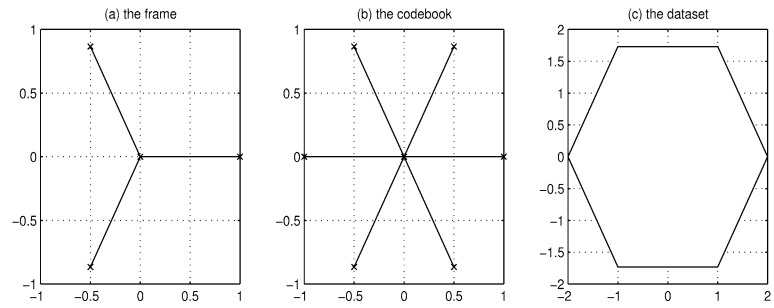

Fig. 2. (a) frame, (b) codebook, (c) the set P.

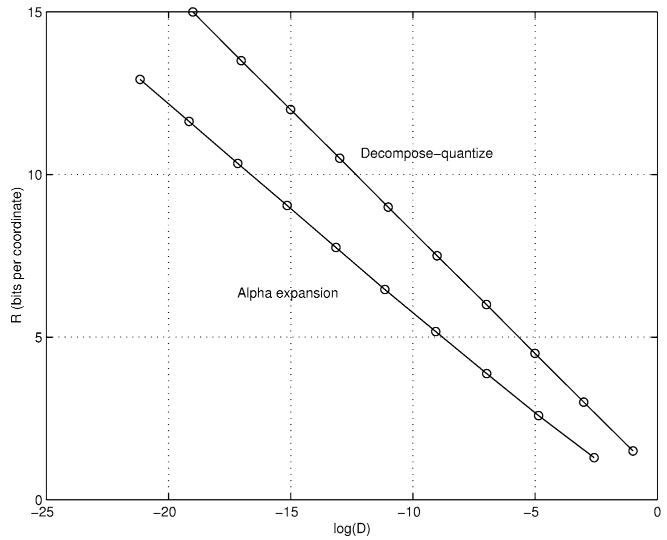

Fig. 3. The $(R, \log_2(D))$ pairs obtained in the experiments.

Example 4.4. Let $\mathcal{F} = \{(1,0), (-\frac{1}{2}, \frac{\sqrt{3}}{2}), (-\frac{1}{2}, -\frac{\sqrt{3}}{2})\}$. Then the associated codebook $\mathcal{C}$ is

$$\mathcal{C} = \left\{(1,0), \left(\frac{1}{2}, \frac{\sqrt{3}}{2}\right), \left(-\frac{1}{2}, \frac{\sqrt{3}}{2}\right), (-1,0), \left(-\frac{1}{2}, -\frac{\sqrt{3}}{2}\right), \left(\frac{1}{2}, -\frac{\sqrt{3}}{2}\right)\right\},$$

and the convex hull $P(\alpha, \mathcal{C})$ is an hexagon. For this codebook, choosing $\alpha = \frac{1}{2}$ we obtain $\Lambda(\alpha, \mathcal{C}) = P(\alpha, \mathcal{C})$ (see Fig. 2).

Since $p = 3$, we have that $\log_2(2p)/p \log_2(\frac{1}{\alpha}) = \log_2(6)$, and hence inequality (11) is strict. Therefore the α-expansion over $\mathcal{C}$ is better than the decompose-quantize procedure for high rate coding. This fact can be confirmed in Fig. 3, where we show the experimental $R_1(D)$ and $R_2(D)$. We have assumed in the experiments that the vector $\mathbf{x}$ is uniformly distributed on P.

5. Conclusions

In this article, we have proposed a class of frame expansions called α-expansions. We have shown convergence results and estimated its rate-distortion caractheristics. We have also theoretically shown that, for high rate coding, it is more efficient than a general decompose-quantize procedure.

In terms of computational complexity, the α-expansions can be very expensive. However, there are some algebraically structured codebooks which admit fast calculations (see [3]).

It is also worthwhile to mention the strong links between the α-expansions and the matching pursuits algorithm, which is a popular frame decomposition algorithm. In the matching pursuits algorithm, the input vector is orthogonally projected on the nearest frame vector, and the same process is repeated recursively for the residuals thus generated. The value of each projection is quantized, being encoded along with the index of the corresponding vector in the codebook. On the other hand, in the α-expansion algorithm, the coefficients of the expansion are of the form α^j, and therefore only the sequence of vector indexes must be encoded.

The matching pursuits is not a decompose-quantize algorithm of the type described in Section 4. In fact, in the matching pursuits, the frame coefficients are not uniformly quantized. The analysis of the rate-distortion characteristics of the matching pursuits algorithm is very difficult, and only partial results are available [6,8]. Hence, a theoretical comparison between the α-expansions and the matching pursuits algorithm was not possible. An interesting issue for further investigation would be to make numerical and theoretical comparisons between both algorithms.

References

[1] C. Bandt, Self-similar sets 5. Integer matrices and fractal tilings of $\mathbf{R}^N$, Proc. Amer. Math. Soc. 112 (2) (1991) 549–562.

[2] M.F. Barnsley, Fractals Everywhere, Academic Press, San Diego, CA, 1988, p. 82.

[3] J.H. Conway, N.J.A. Sloane, Sphere Packings, Lattices and Groups, Springer, 1988.

[4] M. Craizer, E.A.B. da Silva, E.G. Ramos, Convergent algorithms for successive approximation vector quantization with applications to wavelet image compression, IEE Proc. Vision Image Signal Process. 146 (3) (1999) 159–164.

[5] M. Craizer, E.A.B. da Silva, D.A. Fonini Jr., Quantized frame decomposition, in: A. Cohen, C. Rabut, L. Schumaker (Eds.), Curve and Surface Fitting: Saint-Malo 1999, Vanderbilt Press, 2000, pp. 153–160.

[6] G.M. Davis, S. Mallat, M. Avellaneda, Greedy adaptive approximations, J. Constr. Approx. 13 (1997) 57–98.

[7] J. Eckhoff, Helly, Radon, and Carathéodory type theorems, in: P.M. Gruber, J.M. Wills (Eds.), Handbook of Convex Geometry, Ch. 2.1.

[8] V.K. Goyal, Quantized overcomplete expansions: Analysis, synthesis and algorithms, Technical report of the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, 1995.

[9] V.K. Goyal, M. Vetterli, N.T. Thao, Quantized overcomplete expansions in $\mathbf{R}^N$: Analysis, synthesis and algorithms, IEEE Trans. Inform. Theory 44 (1) (1998) 16–31.

[10] A. Gersho, R.M. Gray, Vector Quantization and Signal Compression, Kluwer Academic, 1991.

[11] J.E. Hutchinson, Fractals and self similarity, Indiana J. Math. 30 (5) (1981) 713–747.

[12] S. Mallat, A Wavelet Tour of Signal Processing, Academic Press, San Diego, CA, 1997.

[13] E.A.B. da Silva, D.G. Sampson, M. Ghanbari, A successive approximation vector quantizer for wavelet transform image coding, IEEE Trans. Image Process. 5 (1996) 299–310, Special Issue on Vector Quantization.

[14] N.T. Thao, M. Vetterli, Deterministic analysis of oversampled a/d conversion and decoding improvement based on consistent estimates, IEEE Trans. Signal Process. 42 (1994) 519–531.

[15] N.T. Thao, M. Vetterli, Lower bound on the mean squared error in over-sampled quantization of periodic signals using vector quantization analysis, IEEE Trans. Inform. Theory 42 (2) (1996) 469–479.

[16] P.L. Zador, Development and evaluation of procedures for quantizing multivariate distributions, Ph.D. dissertation, Stanford University, 1963.

[17] P.L. Zador, Asymptotic quantization error of continuous signals and the quantization dimension, IEEE Trans. Inform. Theory 28 (2) (1982) 139–149.