# Geração de Dicionários para Quantização por Planos de Bits Vetoriais

Lisandro Lovisolo e 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
{lisandro, eduardo}@lps.ufrj.br

*Sumário*— ***Em muitas aplicações de quantização vetorial necessitamos de dicionários compostos por vetores uniformemente distribuídos em uma hiperesfera. Uma opção comum nestes casos é dada por dicionários gerados a partir de camadas de reticulados regulares. Porém, estes dicionários são limitados em relação ao número de vetores $K$ e dimensão $N$. Neste artigo descrevemos um novo método para gerar dicionários deste tipo com um número $K$ de vetores e dimensão $N$ arbitrários. Geramos dicionários para pares ($K$,$N$) e comparamos seu desempenho no contexto de Quantização Vetorial por Aproximações Sucessivas (SA-VQ), com os dicionários formados a partir de reticulados regulares, soluções dos problemas de "covering" e "packing" na hiperesfera.***

**Palavras Chave:** *compressão, codificação, quantização vetorial, projeto de dicionários, aproximações sucessivas, reticulados regulares.*

## 1 Introdução

Codificação utilizando planos de bits representa o estado da arte em esquemas de compressão de imagens. O novo padrão JPEG-2000 utiliza codificação através de planos de bits[1]. Em [2, 3] o conceito de planos de bits foi generalizado para vetores através da Quantização Vetorial por Aproximações Sucessivas (SA-VQ). Com este esquema algumas melhoras no desempenho de codificadores foram obtidas. No SA-VQ um vetor $X$ é aproximado por um vetor $X_m$ da seguinte forma:

$$\begin{aligned} &X_m = \sum_{n=1}^{m} \alpha^n Y_{i_n} \\ &\text{onde } Y_{i_n} \in C_N = \{Y_1, Y_2, \ldots, Y_K\}, \\ &i_n \in \{1, 2, \ldots, K\} \text{ e } 0 < \alpha < 1 \end{aligned} \tag{1}$$

$C_N$ é o dicionário composto de vetores com norma unitária, $K$ é a cardinalidade (número de vetores do dicionário) e $m$ o número de passos utilizados no processo de quantização.

Este tipo de representação será útil somente se à medida que aumentarmos o número de passos, e portanto o número $m$ de vetores utilizados para representar $X$, o erro da aproximação $e_m = \|X - X_m\|$ tender a zero. Em [2] é demonstrado que uma condição suficiente para tal é:

$$\begin{cases} \alpha \geq [2\cos\Theta(C_N)]^{-1}\,, & \text{para } \Theta(C_N) \leq \frac{\pi}{4} \\ \alpha \geq \sin\Theta(C_N)\,, & \text{para } \Theta(C_N) \geq \frac{\pi}{4} \end{cases} \tag{2}$$

onde $\Theta(C_N)$ é o ângulo máximo entre qualquer vetor no espaço $\mathbb{R}^N$ e o vetor mais próximo do dicionário $C_N$.

A equação 2 nos indica uma característica que "bons" dicionários para SA-VQ devem ter. Ainda em [2] é demonstrado que o erro na representação $e_m = \|X - X_m\|$ é limitado por $e_m \leq B\alpha^m$. Da equação 2, quanto menor $\Theta(C_N)$ menor o $\alpha$ que garante a convergência do algorítmo SA-VQ e em conseqüência menor o erro. Vê-se que há uma relação entre $\alpha$ e o erro na representação, e por conseqüinte entre $\Theta(C_N)$ e o erro.

Conclui-se então que bons dicionários deverão ter $\Theta(C_N)$ o menor possível. Temos duas maneiras de obter isto:

**1)** Aumentando o número de vetores $K$ que compõem o dicionário $C_N$. Entretanto esta abordagem aumentaria a taxa $R = \frac{m \log_2 K}{N}$.

**2)** Minimizando $\Theta(C_N)$. Para isto devemos distribuir os vetores que compõem o dicionário o mais uniformemente possível sobre a superfície de uma hiperesfera de dimensão $N$. Isto manteria $K$ e $N$ fixos, mantendo a mesma taxa.

Logo, projetar "bons" dicionários para SA-VQ é similar a distribuir uniformente $K$ pontos sobre a superfície de uma esfera de dimensão $N$. Para uma boa discussão sobre o conceito de uniformidade neste caso ver [4]. Este é um problema estudado há séculos [4] porém ainda sem solução geral.

Para o propósito de distribuir um número de vetores sobre a superfície da hiperesfera podemos pensar em três metodologias:

**a)** a algébrica [5, 6], onde utilizamos dicionários gerados a partir de reticulados que são soluções aos problemas do empacotamento ótimo de esferas ou "packing" e da cobertura ótima de esferas ou "covering". Estes dicionários foram utilizados em [2] no contexto de codificação de imagens através de "wavelets" utilizando planos de bits. Entretanto, não há uma solução geral para o problema de distribuir $K$ pontos numa hiperesfera de dimensão $N$. Hardin e Sloane [6] encontraram soluções para dimensões 3, 4 e 5.

**b)** a abordagem geométrica [4], na qual distribuímos os pontos sobre a superfície da esfera utilizando considerações geométricas. Em [4] boas soluções assintóticas para dimensão 3 foram encontradas, porém não há uma metodologia válida para uma dimensão $N$ qualquer.

**c)** o método estocástico, onde utilizamos um algoritmo tipo LBG [7] para gerar o dicionário desejado. O conjunto de treinamento utilizado é composto de vetores cujas componentes são variáveis aleatórias Gaussianas identicamente distribuidas (IIDG). Como neste caso a função densidade de probabilidade depende só do módulo, normalizando os vetores geramos um conjunto de treinamento cuja função densidade de probabilidade é uniformemente distribuída na superfície da hiperesfera. Esta abordagem é em teoria capaz de gerar bons dicionários para qualquer dimensão $N$. Porém o tamanho do conjunto de treinamento (CT) tem que ser bastante grande até mesmo para valores moderados da dimensão $N$ e do número de vetores $K$ do dicionário final. Isto requer, portanto, uma grande carga computacional o que torna esta abordagem impraticável.

Neste artigo propomos um método que utiliza características tanto do método geométrico quanto do estocástico. Projetamos dicionários iniciais e conjuntos de treinamento através de um método geométrico aproximado e os aplicamos no algorítmo LBG de maneira a melhorar o dicionário inicial. O método geométrico utilizado é descrito a seguir.

## 2 O Método Proposto

Nesta parte apresentamos o método desenvolvido. Primeiro introduzimos o método geométrico proposto (GM) baseado em argumentos heurísticos e aproximações que são assintoticamente válidas. Este procedimento será utilizado para gerar dicionários iniciais (DI) e conjuntos de treinamento (CT) que serão aplicados no algorítmo LBG. Na segunda parte desta seção apresentamos as melhoras obtidas com o algoritmo LBG comparando o método combinado proposto ao método estocástico puro de forma a verificar a sua validade.

### 2.1 O Método Geométrico

Primeiro devemos considerar a representação de um ponto no $\mathbb{R}^N$ sobre a superfície de uma hiperesfera de dimensão $N$ e raio unitário em coordenadas esféricas. Estas são dadas por:

$$\begin{aligned} p =& \{1, \omega_1, \omega_2, ..., \omega_{N-1}\} \\ & \text{onde } 0 \le \omega_i \le \pi \quad \text{para } i \in \{1, \ldots, N-2\} \\ & \text{e } 0 \le \omega_{N-1} \le 2\pi \end{aligned} \tag{3}$$

Podemos obter as coordenadas cartesianas do ponto a partir das esféricas através da transformação:

$$\begin{aligned} x_1 &= \sin\omega_1 \\ x_2 &= \sin\omega_1 \sin\omega_2 \\ &\vdots \\ x_{N-1} &= \sin\omega_1 \sin\omega_2 \ldots \sin\omega_{N-1} \\ x_N &= \sin\omega_1 \sin\omega_2 \ldots \sin\omega_{N-2} \cos\omega_{N-1} \end{aligned} \tag{4}$$

Dada a representação em coordenadas esféricas, o algorítmo proposto se baseia na suposição de que quando o número de vetores $K$ é grande, eles definem uma partição da superfície da hiperesfera de dimensão $N$ em hipercubos idênticos de dimensão $N-1$. Considerando que cada hipercubo de lado $\delta$ é definido por pequenas variações dos ângulos $\Delta\omega_1, \Delta\omega_2, \ldots, \Delta\omega_{N-1}$, temos que:

$$\begin{aligned} \delta &= \Delta\omega_1 \\ \delta &= \Delta\omega_j \prod_{i=1}^{j-1} \sin\omega_i, \quad j = 2, \ldots, N-1 \end{aligned} \tag{5}$$

Dado um ponto $\omega_1, \ldots, \omega_{N-1}$ e $\delta$, os valores de $\Delta\omega_1, \Delta\omega_2, \ldots, \Delta\omega_{N-1}$ seguem das equação 5. Através dos $\Delta\omega_i$ podemos distribuir pontos na superfície da esfera e a partir da equação 4 achar os pontos em coordenadas cartesianas. Então temos o algorítmo para alocação dos pontos a seguir:

**compute** $\Delta\omega_1$ usando a eq. 5
**for** $\omega_1 = \frac{\Delta\omega_1}{2}$ **to** $\pi$ **step** $= \Delta\omega_1$ **do**
  **compute** $\Delta\omega_1$ usando a eq. 5
  **for** $\omega_2 = \frac{\Delta\omega_2}{2}$ **to** $\pi$ **step** $= \Delta\omega_2$ **do**
    $\ddots$
    **compute** $\Delta\omega_{N-1}$ usando a eq. 5
    **for** $\omega_{N-1} = \frac{\Delta\omega_{N-1}}{2}$ **to** $2\pi$ **step** $= \Delta\omega_{N-1}$ **do**
      **for** $i = 1 \ldots N-1$ **do**
        **compute** $x_i = \sin\omega_1 \ldots \sin\omega_{i-1} \sin\omega_i$
      **compute** $x_N = \sin\omega_1 \ldots \sin\omega_{N-2} \cos\omega_{N-1}$

O valor de $\delta$ é calculado considerando que se os $K$ hipercubos, cada um de área $\delta^{N-1}$, fornecem uma partição da superfície da hiperesfera, então:

$$A_N = K\delta^{N-1} \tag{6}$$

A área $A_N$ de uma hiperesfera de dimensão $N$ e raio unitário é dada por [5]:

$$A_N = \frac{N\pi^{N/2}}{(N/2)!}, \qquad N \text{ par} \qquad (7)$$

$$A_N = \frac{N2^N\pi^{(N-1)/2}(\frac{(N-1)}{2}!)}{N!}, \quad N \text{ ímpar} \qquad (8)$$

Como as equações 5 e 6 são válidas só quando o número de pontos $K$ tende a infinito, o valor de $\delta$ da equação 6 pode gerar um número de pontos $K' \neq K$. Então deveremos corrigir o valor de $\delta$ através de $\delta = \delta'(\frac{K'}{K})^{\frac{1}{N-1}}$, e repetir este procedimento iterativamente até que $K' = K$.

O procedimento proposto (GM) é assintoticamente válido ($K$ grande) o que o torna bom para gerar conjuntos de treinamento (CT) para o algoritmo LBG. Assim é contornado o problema da necessidade de um CT muito grande requerido pela abordagem estocástica. Verificamos que o método também é bom para gerar dicionários iniciais (DI). Na figura 1 podemos ver os ângulos entre os vetores alocados pelo método para várias cardinalidades do dicionário em dimensão 3. Vemos que o ângulo médio aproxima-se do máximo assintoticamente. Na figura 2 podemos ver os pontos gerados quando o método aloca $K$=90 pontos em dimensão $N$=3, (90,3).

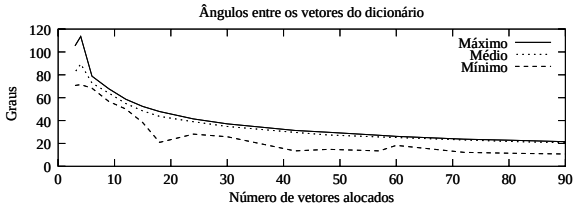


Figura 1: Ângulos máximo médio e mínimo entre os vetores alocados pelo método GM, para $N$=3.

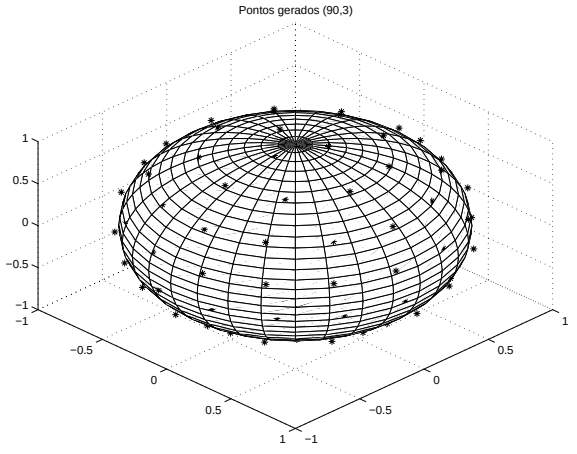


Figura 2: Pontos gerados pelo método para (90,3).

## 2.2 Melhorando os Dicionários

Visto que o método proposto (GM) atinge o objetivo razoavelmente para um número de vetores grande, pode-se utilizá-lo num algorítmo LBG de forma a melhorar dicionários com um pequeno número $K$ de pontos e aumentar a uniformidade. Em nossas simulações a medida de distorção utilizada foi o erro médio quadrático. O algorítmo LBG teve que ser modificado ligeiramente pois o centróide de uma região de Voronoi[7] sobre a hiperesfera não está sobre a hiperesfera. Então para manter o dicionário resultante do LBG sobre a superfície da hiperesfera, a cada iteração normalizamos o centróide de forma a mantê-lo sobre a superfície da mesma.

De maneira a verificar a eficiência do método proposto para gerar dicionários iniciais (DI) e conjuntos de treinamento (CT) avaliamos o desempenho de dicionários gerados pelo LBG para 4 combinações de DI e CT que são referenciadas como:

- **MM**– Dicionário inicial (DI) e conjunto de treinamento (CT) pelo GM;
- **II**– DI e CT componentes Gaussianas identicamente distribuídas (IIDG);
- **IM**– DI IIDG e CT GM;
- **MI**– DI GM e CT IIDG.

Comparou-se o desempenho em termos de número de iterações utilizadas no LBG, e os ângulos máximo, médio e mínimo entre os vetores do dicionário final. Duas coisas foram observadas:

**i)** As simulações **MM** geram dicionários mais uniformes, com um menor ângulo máximo, e maiores ângulos médio e mínimo; por exemplo para $K$=12 e $N$=3 ((12,3)) e CT composto de 9600 vetores, os ângulos máximo, médio e mínimo são repectivamente: 63.28º, 63.07º, 62.64º para **MM**; 61.79º, 60.33º, 58.09º para **MI**; 63.29º, 63.01º, 62.82º para **IM** e 62.07º, 60.36º, 58.16º para **II**.

**ii)** as simulações do tipo **MM** convergem mais rápido para o dicionário final. Isto pode ser observado pelo número de iterações utilizadas no LBG: o tipo **II** necessita 114% iterações a mais que o **MM**, o **IM** 20% e o **MI** 68%.

Destas observações o tipo **MM** é a melhor escolha. Observa-se ainda que a escolha do CT é mais crítica em relação à qualidade do dicionário final enquanto o DI influencia a velocidade do processo.

Podemos observar na figura 3 o processo de convergência do LBG para uma simulação do tipo **MM**, com CTs de diferentes tamanhos para um dicionário (24,4). Vemos claramente que o método possui um bom desempenho.

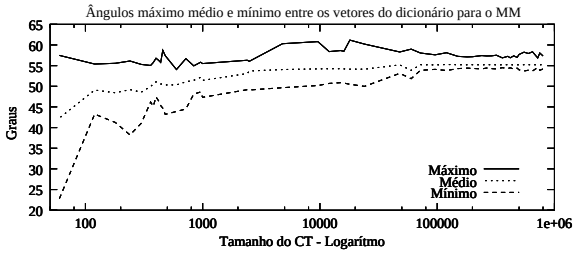


Figura 3: Ângulos máximo médio e mínimo para simulações **MM** para um dicionário (24,4), para diferentes números de vetores no CT.

## 3 Desempenho dos Dicionários

Avaliamos os dicionários gerados no contexto do algorítmo SA-VQ, equação 1, bem como no contexto do algorítmo SA-W-VQ [8]. Na avaliação comparamos o desempenho dos dicionários gerados com os formados a partir das soluções do "covering" e do "packing".

O conjunto de vetores a ser quantizado é formado de vetores com uma distribuição estatisticamente uniforme sobre a superfície da esfera, como descrito antes, e com módulo unitário. Cada dicionário foi avaliado através do erro médio quadrático (**mse**) após $m$ passos. Geramos dicionários em dimensão 3, 4 e 5, e comparamos seus resultados aos dicionários extraídos de [6]. Na figura 4 comparamos o **mse** para o dicionário (24,4) de [6] com o do dicionário (24,4) gerado pelo método **MM**, para $\alpha \in [0.55, 0.7]$ usando 8 passos no algoritmo SA-VQ. Vê-se que o dicionário gerado possui um desempenho melhor que o de [6] para a maior parte dos valores de $\alpha$.

Na tabela 1 temos a comparação de alguns dicionários extraídos de [6], soluções para os problemas do "covering" e "packing" [5] para vários $K$, com $N \in \{3, 4, 5\}$, com os gerados pelo método **MM**. O **mse** mostrado é para o $\alpha$ para o qual é mínimo. Para cada par ($K$,$N$) quantizamos a fonte através do SA-VQ. Vemos que o método proposto gera de forma consistente bons dicionários

Na figura 5 podemos observar a diferença de desempenho em termos de **mse** entre dicionários (24,4), o de [6] e o gerado pelo método **MM**, no contexto de SA-VQ. Nesta vê-se a diferença de desempenho para vários $\alpha \in [0.50, 0.90]$ e números de passos $m \in \{3 \ldots 8\}$. Alguns pontos podem ser observados: **i**) a diferença de desempenho é muito pequena; **ii**) na região de melhor desempenho, $\alpha \in [0.50, 0.60]$, a diferença é nula. Com isto vê-se que os dicionários gerados são consistentes $\forall\ \alpha$ e $\forall\ m$.

De forma a verificar a validade dos dicionários gerados em aplicações que utilizam SA-VQ em esquemas mais elaborados de compressão, utilizamos os dicionários gerados no esquema de compressão de imagens "Successive Approximation Wavelet Vector Quantization" (SA-W-VQ) [8]. O SA-W-VQ codifica a transformada wavelet de uma imagem por planos

| ($K$,$N$) | Geração | passos | $\alpha$ | **mse** |
|---|---|---|---|---|
| (12,3) | "packing" | 9 | 0.59 | 0.000052 |
| (12,3) | método | 9 | 0.58 | **0.000039** |
| (12,3) | "packing" | 6 | 0.59 | 0.001331 |
| (12,3) | "covering" | 6 | 0.58 | 0.001094 |
| (12,3) | método | 6 | 0.58 | **0.001094** |
| (15,3) | "packing" | 6 | 0.57 | 0.000954 |
| (15,3) | "covering" | 6 | 0.56 | 0.000797 |
| (15,3) | método | 6 | 0.56 | **0.000797** |
| (21,3) | "packing" | 6 | 0.55 | 0.000561 |
| (21,3) | "covering" | 6 | 0.55 | 0.000562 |
| (21,3) | método | 6 | 0.55 | **0.000564** |
| (24,4) | "packing" | 6 | 0.59 | 0.001490 |
| (24,4) | método | 6 | 0.60 | **0.001625** |
| (50,4) | "packing" | 6 | 0.56 | 0.000671 |
| (50,4) | método | 6 | 0.50 | **0.000667** |
| (64,5) | "packing" | 6 | 0.59 | 0.001326 |
| (64,5) | método | 6 | 0.59 | **0.001311** |
| (100,5) | "packing" | 6 | 0.57 | 0.000867 |
| (100,5) | método | 6 | 0.57 | **0.000833** |

Tabela 1: Erro médio quadrático para diferentes dicionários.

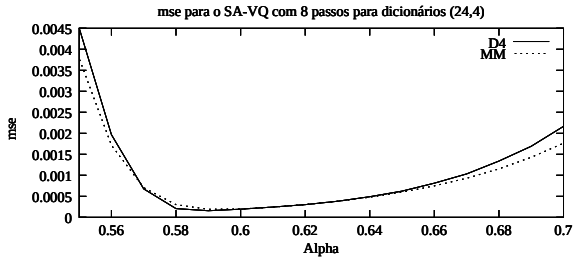


Figura 4: **mse** para o SA-VQ com 8 passos para dicionários (24,4).

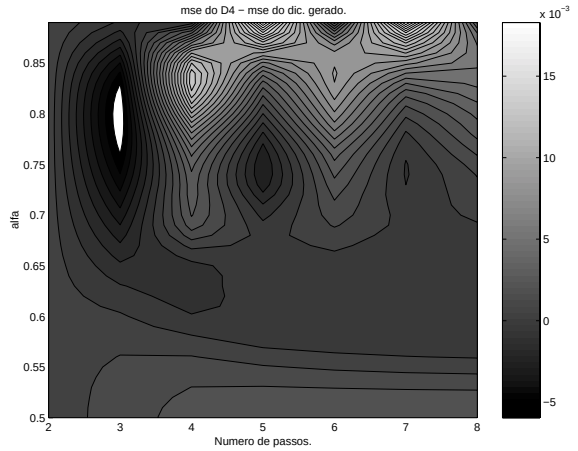


Figura 5: Diferença do **mse** entre o dicionário (24,4) de [6] e o gerado pelo método **MM**.

de bits vetoriais utilizando o conceito de árvores de zeros e codificação aritmética. Codificamos a imagem *Lena* a 0.5 bpp para diferentes valores de $\alpha$ e vários dicionários. Na tabela 2 vemos a comparação do desempenho para o dicionário formado pela primeira camada do reticulado regular $D_4$, melhor "packing" conhecido em dimensão 4 ($D_{4_{S_1}}$), e um dicionário gerado com os mesmos ($K$,$N$), ((24,4)), pelo método proposto, simulação **MM**. Na figura 6 vemos o desempenho dos mesmos dicionários $D_{4_{S_1}}$ e **MM** (24,4) para a faixa de valores possíveis de $\alpha$. Na figura 7, vemos o desempenho do $E_{8_{S_1}}$, dicionário gerado pela primeira camada do $E_8$, comparado com um dicionário gerado com os mesmo ($K$,$N$) do $E_{8_{S_1}}$ (**MM** (240,8)). Pode-se ver que obtemos desempenhos semelhantes, o que corrobora a efetividade do método proposto para a geração de dicionários com $K$ e $N$ arbitrários para SA-VQ.

| PSNR para o SA-W-VQ (dB) | | | |
|---|---|---|---|
| | $\alpha$ | | |
| Dicionário | 0.55 | 0.62 | 0.71 |
| $D_{4_{S_1}}$ | 32.06 | 31.54 | 31.20 |
| **MM** (24,4) | 32.09 | 31.57 | 31.20 |
| $E_{8_{S_1}}$ | 31.46 | 31.72 | 31.22 |
| **MM** (240,8) | 31.02 | 31.62 | 31.16 |

Tabela 2: Comparação entre o PSNR (dB) para o SA-W-VQ para alguns valores de $\alpha$, para o $D_{4_{S_1}}$, o **MM** (24,4), o $E_{8_{S_1}}$ e o **MM** (240,8), para a imagem *Lena* a 0.5 bpp.

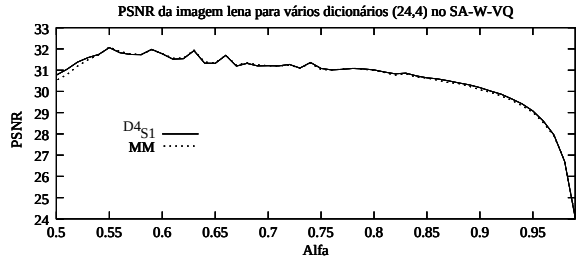


Figura 6: PSNR (db) da imagem *Lena* a 0,5 bpp para o $D_{4_{S_1}}$ e um dicionário com mesmo ($K$,$N$), **MM** (24,4).

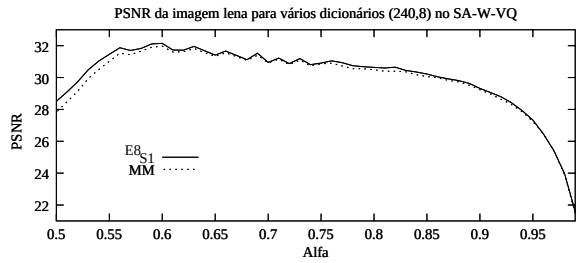


Figura 7: PSNR (db) da imagem *Lena* a 0,5 bpp para o $E_{8_{S_1}}$ e um dicionário com mesmo ($K$,$N$), **MM** (240,8).

## 4 Conclusões

Um método para alocar um número arbitrário de pontos quase uniformemente na superfície de uma hiperesfera foi apresentado. Nos casos em que são conhecidas soluções aos problemas do "packing" e do "covering", vimos que o método proposto gera dicionários tão bons quanto estes. Na verdade $\Theta(C_N)$ mínimo é a solução ao problema do "covering" e os nossos dicionários obtiveram o mesmo desempenho que as soluções conhecidas para este problema em dimensão 3. Vimos também que o desempenho dos dicionários gerados quando aplicados na compressão de imagens é muito semelhante ao dos dicionários normalmente utilizados.

Concluímos então que o método proposto é capaz de gerar bons dicionários de dimensão e cardinalidade arbitrárias. Uma outra aplicação dos dicionários aqui desenvolvidos poderia ser no âmbito da geração de "Frames" [9], para utilização em representações redundantes.

## Referências


[1] ISO/IEC JTC1/SC29/WG1 (ITU-T SG8), *JPEG2000 Verification Model 5.3*, Technical description, JBIG JPEG, November 1999.

[2] CRAIZER, M., da SILVA, E. A. B., RAMOS, E. G., "Convergent Algorithms for Sucessive Approximation Vetor Quantization and Aplications to Wavelet Image Compression", *IEE Procedings - Vision, Image Signal Processing*, v. 146, n. 3, pp. 159–164, June 1999.

[3] da SILVA, E. A. B., CRAIZER, M., "Generalized Bit-Planes for Embedded Codes". *IEEE International Conference on Image Processing*, IEEE, Chicago, September 1998.

[4] SAFF, E. B., KUIJLAARS, A. B. J., "Distributing Many Points on a Sphere", *The Mathematical Intelligencer*, v. 19, n. 1, pp. 5–11, 1997.

[5] CONWAY, J., SLOANE, N., *Sphere Packings Lattices and Groups*, A Series of Comprehensive Studies in Mathematics. 2 ed. New York, New York 10010, USA, Springer Verlag, 1993.

[6] NEIL J. A. SLOANE, R. H. HARDIN, W. D. S., "Neil J. A. Sloane: Home Page", http://www.research.att.com/~njas/, February 2000, And Subpages on Covering and Packing.

[7] GERSHO, A., GRAY, R. M., *Vector Quantization And Signal Compression*, Engineering and Computer Science. 1 ed. Boston/Dordrecht/London, Kluwer Academic Publishers, 1992.

[8] da SILVA, E. A. B., SAMPSON, D. G., GHANBARI, M., "A Successive Approximation Vector Quantizer for Wavelet Transform Image Coding", *IEEE Transactions on Image Processing, Special Issue on Vector Quantization*, v. 5, n. 2, pp. 299–310, February 1996.

[9] GOYAL, V. K., VETTERLI, M., THAO, N. T., "Quantized Overcomplete Expansions in $\mathbb{R}^N$: Analysis, Synthesis, and Algorithms", *IEEE Transactions on Information Theory*, v. 44, n. 1, pp. 16–31, January 1998.