

# Aceleração do Algoritmo *Fuzzy K-means*: Avaliação em Quantização Vetorial de Imagens.

Rodrigo R. Galvão\*\*, Felipe A. B. S. Ferreira\*, F. Madeiro\*\* e Daniel C. Cunha\*\*

\*Instituto de Estudos Avançados de Comunicações, Campina Grande, Paraíba, Brasil

\*\* Escola Politécnica de Pernambuco – Universidade de Pernambuco, Recife, Pernambuco, Brasil  
[rodrigoregisag@gmail.com](mailto:rodrigoregisag@gmail.com), [felipebsferreira@gmail.com](mailto:felipebsferreira@gmail.com), [madeiro@poli.br](mailto:madeiro@poli.br), [dccunha@upe.poli.br](mailto:dccunha@upe.poli.br)

**Abstract** – A method for accelerating the fuzzy k-means algorithm is presented. A performance evaluation of the method is carried out in the scenario of vector quantization codebook design for image compression. Simulation results show that the proposed method leads to a reduction in the number of iterations performed by the fuzzy k-means algorithm. Additionally, it is shown that the convergence speed of the algorithm is increased without sacrificing the quality of the designed codebooks.

**Keywords** – Fuzzy k-means, vector quantization, codebook design, convergence speed.

## 1. Introdução

A quantização vetorial (QV) [1,2], que pode ser vista como uma extensão da quantização escalar em um espaço multidimensional, encontra-se fundamentada na Teoria da Distorção *versus* Taxa, formulada por Shannon, segundo a qual um melhor desempenho é obtido codificando blocos de amostras (isto é, vetores) em vez de amostras individuais (isto é, escalares).

Em diversos sistemas de codificação de voz [3-5] e imagem [6-8], a QV tem sido utilizada com sucesso, permitindo a obtenção de elevadas taxas de compressão. A QV também tem sido utilizada em outras aplicações, tais como, identificação vocal [9,10], esteganografia e marca d'água digital [11], e reconhecimento de palavras manuscritas [12].

Um problema relevante em se tratando de quantização vetorial é o projeto de dicionários. O desempenho de sistemas de processamento de sinais baseados em QV depende fortemente dos dicionários projetados. Dentre as técnicas utilizadas para projeto de dicionários, podem ser citadas: algoritmo *K-means*, algoritmo *Fuzzy K-means* [15] e técnicas de relaxação estocástica. Convém salientar que algoritmos de otimização podem ser usados para melhorar a qualidade de dicionários (quantizadores vetoriais) projetados. Dentre os métodos de otimização, podem ser citados os algoritmos genéticos e os algoritmos meméticos [13].

A principal contribuição deste trabalho é a introdução de uma modificação no algoritmo *Fuzzy K-means*, levada a efeito com o objetivo de aumentar sua velocidade de convergência. São apresentados resultados de simulação envolvendo projeto de dicionários para QV de imagens. Os resultados de simulação mostram que a modificação proposta no presente trabalho pode levar a reduções em até cerca de 30% no número de iterações do algoritmo, ao serem projetados quantizadores vetoriais de tamanho 64 e dimensão 16.

## 2. Quantização Vetorial

A quantização vetorial consiste no mapeamento  $Q$  de um vetor de entrada  $x$  pertencente ao espaço euclidiano  $K$ -dimensional,  $R^k$ , em um vetor pertencente a um subconjunto finito  $W$  de  $R^k$ , ou seja,  $Q: R^k \rightarrow W$ .

O dicionário  $W = \{w_i; i = 1, 2, \dots, N\}$  é o conjunto de vetores de reprodução (também chamados de vetores-código ou vetores de reconstrução),  $k$  é a dimensão do quantizador e  $N$  é o tamanho do dicionário. A taxa de codificação do quantizador vetorial, que mede o número de bits por componente do vetor, é dada por  $\frac{\log_2 N}{k}$ . Em se tratando de codificação de imagens, a taxa é expressa em bits por pixel (*bpp*). Em codificação de forma de onda de voz, é expressa em bit/amostra.

Como a quantização vetorial realiza um mapeamento de padrões de entrada (vetores de entrada  $x$ ) semelhantes em padrões de saída (vetores-código  $w_i$ ) semelhantes, a mesma pode ser vista como uma forma de reconhecimento de padrões, em que um padrão de entrada é “aproximado” por um padrão de referência, pertencente a um conjunto predeterminado (dicionário) de padrões (vetores-código) de referência.

Um problema relevante no âmbito da QV é o projeto de dicionários. O desempenho de sistemas de processamento de sinais baseados em QV depende fortemente da qualidade dos dicionários projetados. De fato, em sistemas de compressão de sinais, a qualidade dos sinais reconstruídos depende dos dicionários; em sistemas de identificação vocal, as taxas de reconhecimento dependem da qualidade dos dicionários.

Neste trabalho, propõe-se uma técnica de aceleração do algoritmo *Fuzzy k-means* aplicado ao projeto de dicionários. São apresentados resultados de simulações envolvendo quantização vetorial de imagens.

### 3. Algoritmo *K-means*

Seja  $X = \{x_i\}, i = 1, 2, \dots, M$  um conjunto de treinamento de vetores  $k$ -dimensionais. O algoritmo *K-means* faz o particionamento do espaço vetorial  $R^k$ , atribuindo cada vetor de treino a um único *cluster* através da busca do vizinho mais próximo. De acordo com a estratégia, o vetor de treino  $x_i$  pertencerá a um único *cluster*. Em outras palavras, o vetor  $x_i$  pertencerá ao *cluster* (ou célula, ou região de Voronoi)  $V(w_j)$  se  $d(x_i, w_j) < d(x_i, w_a) \forall a \neq j$ , em que  $d(x_i, w_j)$  denota a distância euclidiana quadrática entre  $x_i$  e  $w_j$ . Neste caso, diz-se que  $w_j$  é o vizinho mais próximo (VMP) de  $x_i$ . Pode-se associar a busca do vizinho mais próximo a uma função pertinência, assim definida:

$$\mu_j(x_i) = \begin{cases} 1, & \text{se } w_j = \text{VMP}(x_i) \\ 0, & \text{caso contrário.} \end{cases} \quad (1)$$

Desse modo, a medida de distorção, obtida ao se representarem todos os vetores do conjunto de treino pelos respectivos vizinhos mais próximos, é

$$J_1 = \sum_{j=1}^N \sum_{i=1}^M \mu_j(x_i) d(x_i, w_j). \quad (2)$$

Como  $J_1$  é uma função de  $w_j$ , para minimizar a distorção, os vetores  $w_j$  são atualizados de acordo com

$$w_j = \frac{\sum_{i=1}^M \mu_j(x_i) x_i}{\sum_{i=1}^M \mu_j(x_i)}, \forall j = 1, 2, \dots, N. \quad (3)$$

### 4. Algoritmo *Fuzzy K-means*

Diferentemente do algoritmo *K-means*, no algoritmo *Fuzzy K-means* a medida de distorção é dada por [15]

$$J_m = \sum_{j=1}^N \sum_{i=1}^M \mu_j(x_i)^m d(x_i, w_j), \quad (4)$$

onde  $1 < m < \infty$ . Considerando as condições a seguir,

$$\begin{cases} \mu_j(x_i) \in [0,1] \forall i, j, \\ 0 < \sum_{i=1}^M \mu_j(x_i) < M, \\ \sum_{j=1}^N \mu_j(x_i) = 1, \forall i = 1, 2, \dots, M \end{cases}, \quad (5)$$

segue que a minimização da equação (4) permite obter a seguinte expressão para a função de pertinência:

$$\mu_j(x_i) = \frac{1}{\sum_{l=1}^N \left( \frac{d(x_i, w_j)}{d(x_i, w_l)} \right)^{\frac{1}{m-1}}}. \quad (6)$$

Portanto, segundo [15] para um dado conjunto de funções pertinências, os vetores-código evoluirão a cada iteração para minimizar  $J_m$ , de acordo com

$$w_i = \frac{\sum_{j=1}^M \mu_j(x_i)^m x_i}{\sum_{j=1}^M \mu_j(x_i)^m}, \quad \forall j = 1, 2, \dots, N. \quad (7)$$

A nebulosidade nas transições dos *clusters* é controlada pelo parâmetro  $m$ . Quando  $m=1$ , temos os subconjuntos *crisp*. No entanto, à medida que  $m$  aumenta a transição vai ficando nebulosa [15].

### 5. Algoritmo *Fuzzy K-means Acelerado*

Oliveira *et al.*[16] avaliaram alternativas para aceleração do algoritmo *K-means*. Uma das técnicas avaliadas para aumentar a velocidade de convergência (reduzir o número de iterações) do algoritmo consiste em recalculer os vetores-código, ao final de cada iteração do algoritmo de acordo com a expressão a seguir:

$$w_i^{n+1} = w_i^n + s(C(V(w_i^n)) - w_i^n), \quad (8)$$

em que  $s$  é o fator de escala,  $w_i^n$  denota o vetor-código  $w_i$  ao final da  $n$ -ésima iteração e  $C(V(w_i^n))$  é o centróide da partição  $V(w_i^n)$ .

Uma inspeção da Equação (8), proposta por Lee *et al* em [17], revela que a redução do número de iterações do algoritmo *K-means* tem como pressuposto uma tendência de deslocamento dos vetores-código ao longo das iterações segundo uma trajetória aproximadamente retilínea.

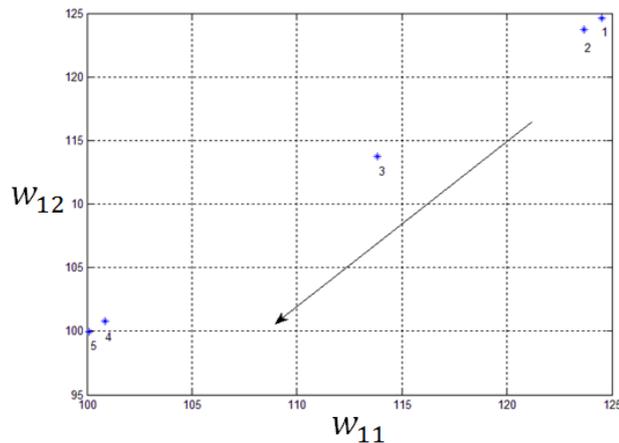
Nas Figuras 1 e 2 estão as atualizações de dois vetores-código a cada iteração do algoritmo *Fuzzy K-means* em um espaço bidimensional, onde o número 1 indica a posição inicial do vetor-código e o demais números (de 2 até 5) indicam as posições ocupadas ao final de cada iteração do algoritmo. Na figura 1,  $w_{11}$  e  $w_{12}$  denotam a primeira e a segunda componente do vetor-código  $w_1$  respectivamente. Na figura 2,  $w_{21}$  e  $w_{22}$  denotam a primeira e a segunda componente do vetor-código  $w_2$  respectivamente. Observa-se em ambas as figuras que o deslocamento do vetor-código é praticamente linear, o que justifica a aplicação da Equação (8) para a redução do número de iterações. Diante desta observação propõe-se no presente trabalho aumentar a velocidade de convergência do algoritmo *Fuzzy K-means* usando o método de *Paliwal e Ramasubramanian* [14], que consiste na atualização do fator de escala  $s$  a cada iteração  $n$ , de acordo com

$$s = 1 + \frac{x}{(x + n)}, \quad (9)$$

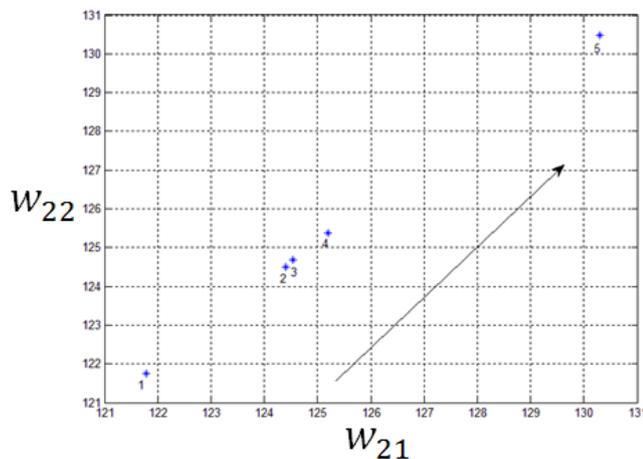
em que  $x > 0$ . Nas simulações realizadas, utilizou-se  $x=9$ .

O fator escala  $s$  proposto em [14] varia inversamente com a iteração  $n$ , satisfazendo a duas condições: 1) a escala deve ser maior que um para assegurar uma convergência mais rápida que a do algoritmo *Fuzzy K-means* convencional; 2) deve ser menor que dois para evitar convergência muito lenta ou não-convergência.

Com o método proposto no presente trabalho, denominado de AFKM (Aceleração do *Fuzzy K-means*), ao final de cada iteração do algoritmo FKM (*Fuzzy K-means* convencional), os vetores-código são recalculados por meio das equações (8) e (9).



**Figura 1** – Atualização de um vetor-código a cada iteração.



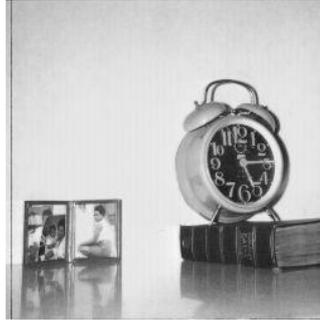
**Figura 2** – Atualização de outro vetor-código a cada iteração.

## 6. Resultados

Os resultados apresentados nesta seção foram obtidos utilizando as imagens *Clock*, *Lena* e *Mandrill* (256 x 256 pixels, com 256 níveis de cinza), apresentadas nas Figuras 3, 4 e 5 respectivamente. Foi considerada QV com dimensão  $k = 16$ , correspondente à utilização de blocos de 4 x 4 pixels. Foram projetados dicionários com  $N = 32, 64, 128, 256$  e 512 vetores-código. Para cada valor de  $N$ , foram utilizadas 20 inicializações diferentes de cada algoritmo. Os algoritmos foram executados tendo como critério de parada um limiar de iteração  $\epsilon = 0,1\%$ . Precisamente, os algoritmos AFKM e FKM param ao final da  $n$ -ésima iteração, se

$$\frac{J_m(n-1) - J_m(n)}{J_m(n)} \leq \epsilon, \quad (10)$$

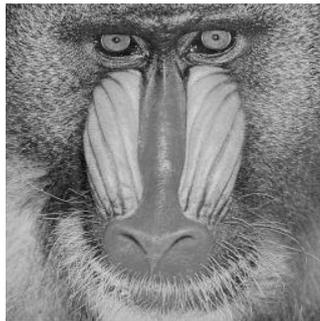
com  $J_m$  dado na Equação (4).



**Figura 3** – Imagem *Clock* original, 256 x 250 pixels, 8,0 bpp.



**Figura 4** – Imagem *Lena* original, 256 x 250 pixels, 8,0 bpp.



**Figura 5** – Imagem *Mandrill* original, 256 x 250 pixels, 8,0 bpp.

Nas simulações, utilizou-se  $m = 5$ , tanto para o algoritmo FKM quanto para o algoritmo AFKM, em conformidade com as recomendações de [15].

A qualidade dos dicionários projetados foi avaliada por meio da relação sinal ruído de pico (PSNR, *Peak signal to noise ratio*), assim definida para imagens 256 x 256, originalmente codificadas a 8,0 bpp:

$$PSNR = 10 \log_{10} \left( \frac{255^2}{\frac{1}{256^2} \sum_{i=1}^{256} \sum_{j=1}^{256} [F(i,j) - \hat{F}(i,j)]^2} \right), \quad (11)$$

em que  $F(i, j)$  e  $\hat{F}(i, j)$  representam os valores de *pixels* das imagens original e reconstruída (quantizada),  $i$  representa a  $i$ -ésima linha e  $j$  denota a  $j$ -ésima coluna de uma imagem (matriz de *pixels*) 256x 256.

São apresentados resultados para: valor médio de relação sinal-ruído de pico (PSNR) das imagens reconstruídas, número médio de iterações, número de casos em que o algoritmo proposto neste trabalho (algoritmo fuzzy k-means acelerado, AFKM) realizou um menor número de iterações (em outras palavras, teve uma maior velocidade de convergência) que a versão original (algoritmo fuzzy k-means convencional, FKM) e número correspondente de casos em que AFKM levou a imagens reconstruídas com um maior valor de PSNR.

Os resultados supramencionados encontram-se organizados na Tabelas 1, 2 e 3, referentes às imagens Clock, Lena e Mandrill respectivamente. Os resultados devem ser entendidos como exemplificado a seguir. Considere, por exemplo, a Tabela 1. Para o conjunto de treino correspondente à imagem Clock, considerando dicionários de tamanho  $N=256$ , o valor médio de PSNR das imagens reconstruídas com dicionários FKM foi 31,18 dB, ao passo que o correspondente valor médio obtido com dicionários AFKM foi 31,33 dB. Vale ressaltar que os valores correspondem a médias obtidas de 20 dicionários projetados (para cada algoritmo, para cada valor de  $N$ , foram utilizados 20 dicionários iniciais distintos). Ainda com relação à Tabela 1, considerando  $N=256$ , o número médio de iterações do algoritmo FKM foi 42,30; o algoritmo AFKM, por sua vez, teve uma maior velocidade média de convergência – de fato, o número médio de iterações para o algoritmo AFKM foi 33,65. Para  $N = 256$ , a Tabela 1 revela, ainda, que, das 20 inicializações consideradas, para 19 o algoritmo AFKM realizou um menor número de iterações; dentre as 19, em 17 os dicionários AFKM levaram a imagens clock reconstruídas com valores de PSNR superiores aos obtidos com os dicionários FKM.

As Tabelas 1, 2 e 3 mostram que, para cada imagem considerada nas simulações, para os diversos valores de  $N$  considerados, o algoritmo AFKM tem, em geral, uma maior velocidade de convergência que a apresentada pelo algoritmo FKM. Em outras palavras, para a grande maioria das 20 inicializações consideradas para cada valor de  $N$ , o algoritmo AFKM realiza um menor número de iterações quando comparado ao algoritmo FKM. Além disso, as tabelas revelam que o aumento de velocidade de convergência não ocorre às custas de um comprometimento de qualidade dos dicionários projetados. Na realidade, o algoritmo AFKM produz, em geral, com um menor número de iterações, dicionários de qualidade similar ou levemente superior (valores de PSNR praticamente iguais ou levemente superiores) à apresentada pelos dicionários FKM. Considere, por exemplo, a Tabela 2. Para  $N=128$ , a técnica proposta no presente trabalho contribuiu para reduzir em cerca de 25% o número de iterações do algoritmo fuzzy k-means. De fato, o algoritmo FKM tem um número médio de iterações igual a 39,75, enquanto o AFKM tem um número médio de iterações igual a 29,40. Para  $N=128$ , os valores médios de PSNR das imagens reconstruídas foram praticamente os mesmos com uso de dicionários FKM e AFKM, respectivamente iguais a 28,85 dB e 28,86 dB.

**Tabela 1** - Resultados obtidos com a imagem *Clock*.

$N$	PSNR Médio		Nº Médio de Iterações		Nº de casos em que AFKM realizou um menor nº de iterações quando comparado a FKM	Nº de casos em que os dicionários AFKM levam a valores de PSNR maiores que as obtidos com os dicionários FKM
	FKM	AFKM	FKM	AFKM		
<b>32</b>	26,61	26,60	27,15	22,20	15	8
<b>64</b>	27,95	28,07	30,20	25,45	14	14
<b>128</b>	29,41	29,56	38,15	31,70	14	13
<b>256</b>	31,18	31,33	42,30	33,65	19	17
<b>512</b>	33,15	33,69	41,85	35,50	17	16

**Tabela 2** - Resultados obtidos com a imagem Lena.

N	PSNR Médio		N° Médio de Iterações		N° de casos em que AFKM realizou um menor n° de iterações quando comparado a FKM	N° de casos em que os dicionários AFKM levam a valores de PSNR maiores que as obtidos com os dicionários FKM
	FKM	AFKM	FKM	AFKM		
<b>32</b>	26,45	26,48	28,10	22,35	17	12
<b>64</b>	27,63	27,63	32,10	22,25	18	10
<b>128</b>	28,85	28,86	39,75	29,40	19	12
<b>256</b>	30,13	30,14	46,45	34,35	17	13
<b>512</b>	31,50	31,68	52,90	39,35	19	18

**Tabela 3** - Resultados obtidos com a imagem Mandrill.

N	PSNR Médio		N° Médio de Iterações		N° de casos em que AFKM realizou um menor n° de iterações quando comparado a FKM	N° de casos em que os dicionários AFKM levam a valores de PSNR maiores que as obtidos com os dicionários FKM
	FKM	AFKM	FKM	AFKM		
<b>32</b>	23,01	23,03	19,85	15,80	19	11
<b>64</b>	23,50	23,52	21,30	16,75	20	15
<b>128</b>	24,05	24,10	24,65	19,90	20	17
<b>256</b>	24,72	24,76	30,50	24,80	20	16
<b>512</b>	25,56	25,64	42,70	35,95	19	18

## 7. Conclusão

Neste trabalho foi apresentado um método para aumentar a velocidade de convergência do algoritmo *fuzzy K-means*, aplicado ao projeto de dicionários para compressão de imagens baseada em quantização vetorial. O método consiste, essencialmente, em recalcular os vetores-código ao final de cada iteração do algoritmo, por meio de um procedimento que tem como pressuposto o deslocamento dos vetores-código segundo trajetórias aproximadamente retilíneas ao longo das iterações do algoritmo. Por meio de simulações envolvendo projeto de dicionários para quantização vetorial de imagens, observou-se que o método proposto contribui para reduzir o número de iterações realizadas pelo algoritmo *fuzzy K-means*. Os resultados de simulações mostraram que o aumento da velocidade de convergência do algoritmo não ocorre às custas de prejuízo de qualidade dos dicionários projetados. Em outras palavras, a relação sinal-ruído de pico (PSNR) das imagens reconstruídas com uso de dicionários projetados pelo método proposto é praticamente a mesma ou ligeiramente superior à PSNR das imagens reconstruídas com uso de dicionários projetados pelo versão original do algoritmo *fuzzy K-means*. Em projetos de dicionários de dimensão 16 e tamanho 256, o método apresentado neste trabalho levou a reduções de até cerca de 25% no número médio de iterações realizadas, considerando o uso de 20 inicializações distintas nas simulações.

## 8. Referências

- [1] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, Boston, 1992.
- [2] R. M. Gray. “Vector Quantization”. *IEEE ASSP Magazine*, pp. 4–29, April 1984.
- [3] K. K. Paliwal and B. S. Atal. “Efficient Vector Quantization of LPC Parameters at 24 Bits/Frame”. *IEEE Transactions on Speech and Audio Processing*, vol. 1, no. 1, pp. 3–14, January 1993.
- [4] Y. Agiomyrgiannakis and Y. Stylianou. “Conditional Vector Quantization for Speech Coding”. *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 2, pp. 377–386, February 2007.
- [5] S. Chatterjee and T. Sreenivas. “Conditional PDF-Based Split Vector Quantization of Wideband LSF Parameters”. *IEEE Signal Processing Letters*, vol. 14, no. 9, pp. 641–644, September 2007.
- [6] K. Sasazaki, S. Saga, J. Maeda and Y. Suzuki. “Vector Quantization of Images with Variable Block Size”. *Applied Soft Computing*, , no. 8, pp. 634–645, 2008.
- [7] A. Averbuch, D. Lazar and M. Israeli. “Image Compression Using Wavelet Transform and Multiresolution Decomposition”. *IEEE Transactions on Image Processing*, vol. 5, no. 1, pp. 4–15, January 1996.
- [8] W. T. A. Lopes, F. Madeiro, B. G. Aguiar Neto and M. S. Alencar. “Combining Modulation Diversity and Index Assignment to Improve Image VQ for a Rayleigh Fading Channel”. *Learning and Nonlinear Models*, vol. 2, no. 1, pp. 22–33, 2004.
- [9] J. He, L. Liu and G. Palm. “A Discriminative Training Algorithm for VQ-based Speaker Identification”. *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 353–356, May 1999.
- [10] F. Madeiro, J. M. Fechine, W. T. A. Lopes, B. G. Aguiar Neto and M. S. Alencar. “Identificação Vocal por Frequência Fundamental, QV e HMMS”. In *Em-TOM-Ação: A Prosódia em Perspectiva*, edited by M. A. M. Aguiar and F. Madeiro, chapter 4, pp. 91–120. Editora Universitária da UFPE, Recife, PE, first edition, 2007.
- [11] W.-C. Chen and M.-S. Wang. “A Fuzzy c-means Clustering-based Fragile Watermarking Scheme for Image Authentication”. *Expert Systems with Applications*, vol. 36, no. 2, pp. 1300–1307, March 2009.
- [12] L. R. Veloso. “Sistema de Reconhecimento de Palavras Manuscritas Dependente do Usuário”. Tese (Doutorado em Engenharia Elétrica), Universidade Federal de Campina Grande, 2009.
- [13] C. R. B. Azevedo, R. A., E. L. Bispo Junior, T. A. E. Ferreira, W. T. A. Lopes and F. Madeiro. “Um Algoritmo Memético para a Otimização de Quantizadores Vetoriais”. *Learning and Nonlinear Models*, vol. 5, no. 1, p. 1-15, 2008.
- [14] K. K. Paliwal and V. Ramasubramanian. (2000). Comments on “Modified K-means algorithm for vector quantizer design”, *IEEE Transactions on Image Processing* 9(11): 1964-1967.
- [15] N. B. Karayiannis and Pin-I Pai, “Fuzzy Vector Quantization Algorithms and Their Application in Image Compression”, *IEEE Transactions on Image Processing*, vol. 4, no. 9, September, 1995.
- [16] F. J. A. Oliveira, J. P. F. C. Freire, A. L.O. Cavalcanti, E. N. Arcoverde, P. C. M. Albuquerque, W. T. A. Lopes e F. Madeiro. Aceleração do Algoritmo *K-means*. Congresso Brasileiro de Automática (CBA), 2004, Gramado, RS, *Anais do XV Congresso Brasileiro de Automática*, 2004.
- [17] D. Lee, S. Baek and K. Sung. Modified K-means algorithm for vector quantizer design, *IEEE Signal Processing Letters* 4(1): 2-4, 1997.