

# Algoritmo FA Modificado Aplicado ao Projeto de Quantizadores Vetoriais

Verusca Severo\*, Herbert A. de Sá Leitão†, Juliano B. Lima‡, Waslon T. A. Lopes §, Francisco Madeiro¶

\*Universidade de Pernambuco-UPE, Recife, Pernambuco

Email: verusca.severo@gmail.com

†Universidade Federal de Pernambuco-UFPE, Recife, Pernambuco

Email: herbert.leitao@ufpe.br

‡Universidade Federal de Pernambuco-UFPE, Recife, Pernambuco

Email: juliano\_bandeira@ieee.org

§Universidade Federal da Paraíba-UFPB, João Pessoa, Paraíba

Email: waslon@ieee.org

¶Universidade de Pernambuco-UPE, Recife, Pernambuco

Email: madeiro@poli.br

**Resumo**—A quantização vetorial (QV) é uma técnica de compressão que alcança elevadas taxas de compressão e tem sido utilizada com sucesso na área de processamento de sinais, como por exemplo em sistemas de esteganografia e marca d'água digital. O desempenho de sistemas de processamento de sinais baseados em QV depende da qualidade dos dicionários projetados, visto que a QV introduz distorção ao se representarem os vetores de entrada pelos respectivos vetores do dicionário. Este trabalho propõe modificações no algoritmo FA-LBG (*firefly algorithm* - Linde-Buzo-Gray), que aplica o algoritmo do vagalume ao projeto de dicionários para QV. A versão modificada difere da original sobretudo por permitir uma maior influência do conjunto de treino no projeto de dicionários. Resultados de simulação mostram que a técnica supera sua versão original e o tradicional algoritmo LBG no que diz respeito à qualidade dos dicionários projetados, avaliada via relação sinal-ruído de pico das imagens reconstruídas.

**Palavras-chave:** Quantização vetorial, compressão de imagens, projeto de dicionários, algoritmo do vagalume.

## I. INTRODUÇÃO

O crescimento da *Internet* proporcionou o aumento significativo do uso de dados em formato digital. Com isso, é importante representar adequadamente os sinais para que se adequem às limitações impostas pelas redes de comunicações [1]–[3]. Com o uso de técnicas de compressão, é possível alcançar esse objetivo.

A quantização vetorial é uma técnica de compressão com perdas. O objetivo da técnica é representar uma sequência de vetores de entrada de dimensão  $K$  por uma sequência de vetores de mesma dimensão, de maneira a reduzir a quantidade de bits necessária para se representar a sequência de vetores de entrada [4] [5]. O processo de quantização vetorial de um sinal de entrada pertencendo ao espaço euclidiano  $K$ -dimensional se dá através do mapeamento, segundo um critério de distância mínima, de cada vetor deste sinal de entrada em um vetor pertencente a um subconjunto finito  $W$  de  $\mathbb{R}^K$  formado por  $N$  elementos denominados vetores-código,  $W = \{w_n, n = 1, 2, 3, \dots, N\}$ . Esse subconjunto é chamado dicionário [4], [5]. O dicionário apresenta forte influência no

desempenho de sistemas de processamento de sinais baseados em quantização vetorial, tais como sistemas de esteganografia [6]–[8] e marca d'água digital [9]–[11].

O projeto de dicionários busca encontrar  $N$  vetores-código para um determinado conjunto de  $M$  vetores de treino que reduza a distorção introduzida ao se representarem os vetores de entrada  $x$  por suas correspondentes versões quantizadas. O projeto corresponde a um processo de *clusterização* ou agrupamento de dados que consiste em particionar ou dividir o conjunto de treino em  $M$  vetores e agrupa-los em  $N$  regiões ou *clusters* de modo que vetores da mesma região apresentem alta similaridade entre si e apresentem baixa similaridade com os vetores das outras regiões. O estudo de técnicas para geração de dicionários mais representativos é um tema de pesquisa bastante explorado [12]–[15]. Dentre as diversas técnicas destinadas à produção de dicionários, o algoritmo LBG (Linde, Buzo e Gray) [16] se destaca por ser o mais popular e mais utilizado para o projeto de dicionários. Algoritmos de computação bioinspirada tais como em [12], [13], [15], [17] tem sido propostos como alternativas ao algoritmo LBG, melhorando o processo de quantização vetorial.

Horng propôs um método baseado no algoritmo do vagalume (FA) para construção do dicionário de quantização vetorial [13]. Na natureza, os vagalumes utilizam seu brilho para atrair um parceiro para o ritual de acasalamento, alertar contra predadores e para atrair presas [18]. No algoritmo FA, o brilho do vagalume é associado a uma função objetivo a ser otimizada, que no caso do método proposto por Horng corresponde ao inverso da distorção entre os vetores do conjunto de treino e os respectivos vetores do dicionário. A posição de cada vagalume corresponde a uma solução candidata. O método utiliza o resultado do algoritmo LBG como uma das soluções candidatas iniciais, enquanto as demais soluções correspondem a dicionários gerados aleatoriamente a partir de um conjunto de treino. Cada vagalume no espaço de busca é atraído por um vagalume de maior brilho que ele e, dessa forma, novas soluções são geradas até se encontrar o vagalume que maximize a função objetivo.

Este trabalho apresenta modificações no algoritmo do vaga-

lume proposto por Horng em [13] com o objetivo de melhorar a qualidade dos dicionários projetados. As modificações dizem respeito à forma de inicialização, à etapa de treinamento e à definição do critério de parada do algoritmo. A versão modificada é inicializada com um conjunto de dicionários iniciais aleatórios que são obtidos a partir de um conjunto de treino. Cada dicionário representa uma solução e o objetivo do método é encontrar a melhor solução dentre o conjunto de possíveis soluções. Quanto à etapa de treinamento, a versão modificada introduz na sua versão original a etapa de atualização dos vetores-código através do cálculo dos centroides utilizado pelo algoritmo LBG, com o intuito de permitir maior influência do conjunto de treino no projeto de dicionário. Como critério de parada, é adotado o teste de convergência definido pelo algoritmo LBG, que monitora a distorção a cada iteração e encerra o treinamento do dicionário quando um limiar de distorção pré-definido é estabelecido.

O restante deste artigo encontra-se organizado da seguinte forma: a Seção II aborda o algoritmo LBG para o projeto de dicionários para quantização vetorial, a Seção III apresenta o algoritmo do vagalume, a Seção IV descreve o funcionamento do FA-LBG, a Seção V descreve o funcionamento do FA-LBG modificado, os resultados são apresentados na Seção VI e as conclusões na Seção VII.

## II. ALGORITMO LBG

O algoritmo LBG, também conhecido como algoritmo de Lloyd generalizado (GLA, *Generalized Lloyd Algorithm*) é um algoritmo de agrupamento ou *clusterização*, que começa com uma solução inicial que é melhorada de forma iterativa visando satisfazer as condições de centróide e de vizinho mais próximo. O algoritmo LBG é uma generalização do algoritmo proposto por Lloyd [19] em 1957 para quantização escalar, ou seja, para dados em uma dimensão. Em 1980, Linde, Buzo e Gray [16] estenderam o algoritmo de Lloyd para ser aplicado a mais de uma dimensão, ou seja, para vetores de dimensão  $K$ , daí o LBG também ser conhecido como  $K$ -means ou GLA.

Inicialmente, seja o conjunto de treino  $X = \{x_m, m = 1, 2, 3, \dots, M\}$  formado por  $M$  vetores de dimensão  $K$  pertencentes ao espaço vetorial  $\mathbb{R}^K$ . O algoritmo LBG particiona o espaço  $\mathbb{R}^K$  em  $N$  regiões de Voronoi  $S_n$ , formando o dicionário inicial com  $N$  vetores-código. Cada região é representada por um único vetor-código  $w_n$  e, a partir da busca do vizinho mais próximo, que consiste em selecionar o vetor-código que apresente maior semelhança segundo uma medida de distância, atribui cada vetor treino  $x_m$  a uma única região de Voronoi, de acordo com

$$S_n = \{x_m \mid d(x_m, w_n) < d(x_m, w_o) \forall o \neq n\}, \quad (1)$$

em que  $d(\cdot)$  denota uma medida de distância, por exemplo, a distância euclidiana quadrática, definida por

$$d(x_m, w_n) = \sum_{j=1}^K (x_{mj} - w_{nj})^2, \quad (2)$$

em que  $x_{mj}$  é a  $j$ -ésima componente de  $x_m$  e  $w_{nj}$  é a  $j$ -ésima componente do vetor-código  $w_n$ .

Com todos os vetores do conjunto treino representados pelos seus respectivos vizinhos mais próximos, pode-se calcular a distorção por

$$D_t = \sum_{n=1}^N \sum_{x_m \in S_n} d(x_m, w_n), \quad (3)$$

em que  $t$  corresponde a  $t$ -ésima iteração.

Com o objetivo de minimizar a distorção, os vetores-código são calculados e atualizados como os centroides das regiões de Voronoi, a partir do cálculo da média aritmética das  $j$ -ésimas componentes de todos os vetores código pertencentes à região  $S_n$  de acordo com

$$w_{nj} = \frac{1}{M_n} \sum_{x_m \in S_n} x_{mj}, \quad (4)$$

em que  $M_n$  representa a quantidade de vetores de treino alocados na região  $S_n$  e  $x_{mj}$  representa a  $j$ -ésima componente de  $x_m$ .

A alocação dos vetores de treino nas regiões de Voronoi e atualização dos vetores-código são realizadas até que o critério de parada seja atingido. O critério de parada é baseado na distorção introduzida ao se representarem os vetores treino por seus correspondentes vetores-código. A distorção é monitorada a cada iteração e decresce monotonicamente. O critério de parada é atingido, encerrando o processo, quando a queda na distorção total na iteração em questão, relativa à distorção total na iteração anterior, se apresentar abaixo de um determinado limiar de distorção, previamente estabelecido, ou seja, se

$$\frac{(D_{t-1} - D_t)}{D_t} \leq \epsilon, \quad (5)$$

em que  $\epsilon$  corresponde ao limiar de distorção.

Dado que  $t$  é a iteração do algoritmo LBG, este pode ser resumido na seguinte sequência de passos:

- Passo 1: (Inicialização) Defina  $W_0$  como um dicionário inicial e faça  $t = 0$  e  $D_{-1} = \infty$ ;
- Passo 2: (Particionamento) Seja  $W_t$  o dicionário na  $t$ -ésima iteração, aloque cada vetor treino  $x_m$ , na respectiva região de Voronoi segundo a regra do vizinho mais próximo;
- Passo 3: (Cálculo da distorção) Calcule a distorção total, de acordo com a Equação (3).
- Passo 4: (Teste de parada) Se o critério de parada for satisfeito de acordo com a Equação (5), o algoritmo para e retorna  $W_t$  representando o dicionário final; caso contrário, continua;
- Passo 5: (Atualização dos vetores-código) Calcule os novos centróides de cada região de acordo com a Equação (4), atualizando os vetores-código. Faça  $t = t + 1$  e retorne ao Passo 2.

## III. ALGORITMO DO VAGALUME

O algoritmo do vagalume foi proposto por Yang em 2008 [20] e é inspirado no fenômeno da bioluminescência dos

vagalumes, que é utilizado no ritual do acasalamento, para atrair presas e deter possíveis predadores [18].

Para o funcionamento do algoritmo do vagalume, duas questões precisam ser definidas: como se dá a variação da intensidade da luz percebida pelos vagalumes e como é formulada a atratividade entre os vagalumes [21]. Para simplificar o método, Yang definiu que a atratividade de um vagalume é determinada pela intensidade da luz emitida por ele e que essa intensidade emitida é proporcional à função objetivo.

Seja  $\gamma$  o coeficiente de absorção da luz pelo meio. A intensidade de luz percebida por um vagalume decai em função da distância entre os vagalumes [22] e é dada por

$$d(f_i, f_j) = I_0 e^{-\gamma d^2(f_i, f_j)}, \quad (6)$$

em que  $I_0$  corresponde à intensidade da luz emitida pelo vagalume e  $d(f_i, f_j)$  é a distância euclidiana entre quaisquer dois vagalumes,  $f_i$  e  $f_j$ , e é calculada de acordo com

$$d(f_i, f_j) = \sum_{k=1}^K (f_{ik} - f_{jk})^2, \quad (7)$$

em que  $K$  representa a dimensão do problema e  $f_{ik}$  a  $k$ -ésima componente do  $i$ -ésimo vagalume.

Segundo Yang, a atratividade de um vagalume pode ser formulada como

$$\beta(d(f_i, f_j)) = \beta_0 e^{-\gamma d^2(f_i, f_j)}, \quad (8)$$

em que  $\beta_0$  corresponde à atratividade do vagalume para  $d(f_i, f_j) = 0$ .

De acordo com Yang, um vagalume  $i$  é atraído por um vagalume  $j$  de maior brilho, e seu movimento em um determinado tempo  $t$  é definido por

$$f_i^t = f_i^{(t-1)} + \beta(f_j^{(t-1)} - f_i^{(t-1)}) + \alpha(rand - 1/2). \quad (9)$$

A Equação 9 consiste em três termos [23]:

1º termo:  $f_i^{(t-1)}$ , que corresponde à posição atual do vagalume e representa uma possível solução ao problema.

2º termo:  $\beta(f_j^{(t-1)} - f_i^{(t-1)})$ , que corresponde à atração por outro vagalume de maior brilho.

3º termo:  $\alpha(rand - 1/2)$ , que corresponde ao movimento aleatório do vagalume, em que o parâmetro  $\alpha$  regula a inserção de aleatoriedade no caminho percorrido pelo vagalume ( $rand$  representa um número aleatório entre 0 e 1).

Caso não exista para  $f_i$  um vagalume de maior brilho,  $f_i$  se moverá de maneira aleatória segundo a Equação 10.

$$f_i^t = f_i^{(t-1)} + \alpha(rand - 1/2). \quad (10)$$

A Figura 1 ilustra o fluxograma de funcionamento do algoritmo do vagalume.

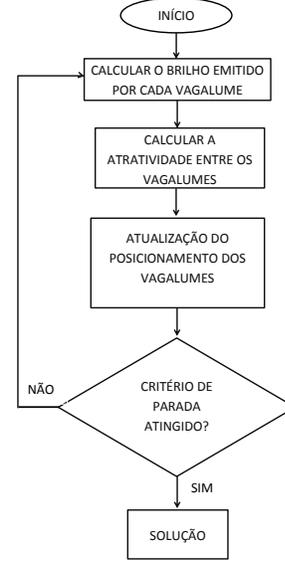


Figura 1. Fluxograma do Algoritmo FA.

Dada uma população inicial de  $F$  vagalumes  $f_i$  ( $i = 1, 2, 3, \dots, F$ ), em que cada vagalume representa um candidato à solução, o algoritmo inicia com o cálculo do brilho emitido por cada vagalume que definirá o comportamento do movimento dos vagalumes (quem atrai e quem será atraído), em seguida, determina a atratividade entre os vagalumes de acordo com a Equação (8) e move cada vagalume  $f_i$  em direção a outro vagalume de maior brilho segundo a Equação (9). Se não existir outro vagalume que apresente maior brilho que o vagalume em questão, este moverá de forma aleatória de acordo com a Equação (10). Esse processo é repetido até que o critério de parada seja estabelecido.

#### IV. ALGORITMO FA-LBG

O algoritmo FA-LBG proposto por Horng em [13] aplica o algoritmo do vagalume ao projeto de dicionário para quantização vetorial.

No FA-LBG cada vagalume representa um dicionário de tamanho  $N$  e dimensão  $K$ . Cada vagalume corresponde a uma possível solução para o problema. A função *fitness*, que avalia o potencial de cada solução, é definida no FA-LBG como o inverso da distorção, como segue:

$$fitness(f_a) = \frac{1}{D(f_a)}. \quad (11)$$

O objetivo do FA-LBG é encontrar um dicionário que maximize a função *fitness* para os vetores de entrada.

O algoritmo FA-LBG é inicializado com  $F$  vagalumes. Um dos  $F$  vagalumes é o resultado do algoritmo LBG, enquanto os  $F - 1$  vagalumes são gerados aleatoriamente. O algoritmo pode ser resumido na seguinte sequência de passos:

PASSO 1: (Inicialização e definição dos parâmetros) Inicialize o algoritmo com  $F$  vagalumes, sendo um vagalume o resultado do algoritmo LBG e os  $F - 1$  vagalumes gerados

aleatoriamente e defina os parâmetros  $\alpha$ ,  $\beta_0$ ,  $\gamma$  e número de ciclos.

PASSO 2: (Busca pelo vagalume de maior brilho) Calcule o *fitness* de cada vagalume e determine o de maior brilho.

PASSO 3: (Atualização do posicionamento dos vagalumes, exceto o de maior brilho) Para cada vagalume, exceto o que apresenta maior brilho, selecione aleatoriamente outro vagalume de maior brilho para seguir e atualize sua posição de acordo com a Equação 9.

PASSO 4: (Critério de Parada) Se o número de ciclos for atingido, o algoritmo é finalizado com a saída sendo o vagalume de maior brilho. Caso contrário, atualize a posição do vagalume de maior brilho de acordo com a Equação 10 e retorne ao PASSO 2.

## V. ALGORITMO FA-LBG MODIFICADO

O algoritmo FA-LBG modificado foi obtido a partir de alterações realizadas no FA-LBG, permitindo uma maior influência do conjunto de treino no projeto de dicionários. As alterações dizem respeito à forma de inicialização, à etapa de treinamento e à definição do critério de parada do algoritmo.

O FA-LBG modificado realiza as seguintes alterações no algoritmo FA-LBG:

Etapa de inicialização: Cada vagalume corresponde a um dicionário aleatório inicial, obtido do conjunto de treino.

Etapa de treinamento: Com o objetivo de permitir uma maior influência do conjunto de treino no projeto de dicionário, foi introduzido à etapa de atualização do dicionário (que na versão original restringe-se ao reposicionamento dos vagalumes) o cálculo dos centroides do algoritmo LBG.

Critério de parada: Como teste de convergência do algoritmo, utiliza-se o critério de parada do algoritmo LBG, que depende de um limiar de distorção previamente definido.

No FA-LBG, o dicionário é iterativamente atualizado através do reposicionamento dos vagalumes. A versão modificada do FA-LBG realiza o cálculo dos centroides de cada região de Voronoi e o reposicionamento dos vagalumes. Dessa forma, a atualização do dicionário acontece em duas etapas:

- primeiro, com a atualização dos novos centroides das regiões de Voronoi, segundo a Equação 4;
- e em seguida, com a atualização da posição dos vagalumes, segundo a Equação 9.

Como o cálculo do centroide de cada região de Voronoi corresponde à média aritmética das componentes de todos os vetores de treino pertencentes a região de Voronoi em questão, o objetivo de introduzir o cálculo dos centroides na etapa de atualização do dicionário no FA-LBG modificado é permitir uma maior influência do conjunto de treino no projeto de dicionários buscando minimizar a distorção introduzida ao se representarem os vetores de treino pelos respectivos vetores-código.

O critério de parada do algoritmo FA-LBG modificado é definido em função da distorção monitorada. A quantização vetorial introduz distorção ao se representarem os vetores de

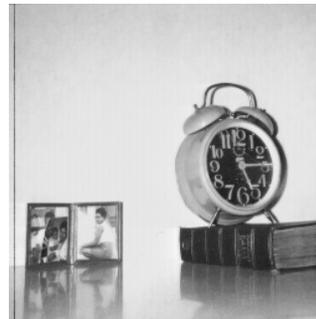
treino pelos seus vizinhos mais próximos. No algoritmo LBG, essa distorção introduzida é monitorada a cada iteração e serve de base para o critério de parada do algoritmo. O algoritmo FA-LBG modificado utiliza a mesma regra de parada do LBG, que define o fim do treinamento do dicionário quando a queda na distorção total em determinada iteração relativa à queda na distorção total na iteração anterior a iteração em questão estiver abaixo de um determinado limiar de distorção.

## VI. RESULTADOS

Os resultados apresentados nesta seção foram obtidos para dicionários projetados utilizando como conjunto de treino as imagens Boat, Clock e Lena, todas de dimensão  $256 \times 256$  pixels, no formato *Portable Gray Map* (PGM), codificadas a 8 bits por pixel (*bpp*) (256 níveis de cinza), obtidas da base de imagens do *Signal and Image Processing Institute* (SIPI) da *University of Southern California* (USC) [24]. A Figura 2 apresenta as imagens utilizadas.



(a) Boat.



(b) Clock.



(c) Lena.

Figura 2. Imagens utilizadas nas simulações.

Os dicionários foram projetados com dimensão ( $K$ ) igual a 16 (blocos  $4 \times 4$  pixels) e tamanho ( $N$ ) igual a 32, 64, 128, 256 e 512 vetores-código, correspondentes às respectivas taxas de codificação ( $R$ ) iguais a 0,3125 *bpp*, 0,375 *bpp*, 0,4375 *bpp*, 0,5 *bpp* e 0,5625 *bpp*. Quanto à etapa de inicialização dos dicionários iniciais, esta foi realizada a partir da escolha aleatória de blocos de pixels da imagem treino.

Cada dicionário inicial é representado por um vagalume. No algoritmo FA-LBG, foram utilizados  $F = 10$  vagalumes, sendo que um dos vagalumes corresponde a um dicionário projetado pelo algoritmo LBG e os ( $F-1$ ) vagalumes restantes correspondem a dicionários gerados aleatoriamente a partir de uma imagem treino. Na versão modificada do FA-LBG também foram utilizados como ponto de partida  $F =$

10 vagalumes, sendo todos gerados aleatoriamente. Foram utilizados os mesmos ( $F-I$ ) dicionários aleatórios utilizados na versão original do FA-LBG e mais um dicionário gerado aleatoriamente a partir de uma imagem treino, compondo dessa forma  $F$  dicionários iniciais aleatórios.

O algoritmo do vagalume, basicamente, é controlado por três parâmetros:  $\alpha$ ,  $\beta_0$  e  $\gamma$ . O parâmetro  $\alpha$  regula a aleatoriedade no movimento do vagalume. Para se escolher esse parâmetro foram testados valores de  $\alpha$  variando entre 0 e 1. Os parâmetros  $\beta_0$  e  $\gamma$  definem a atratividade dos vagalumes de acordo com a Equação (8). Se  $\gamma$  tender a zero resultará em  $\beta$  ser aproximadamente igual a  $\beta_0$ , fazendo com que a capacidade de atração seja constante em qualquer lugar do espaço de busca, ou seja, independente da distância entre os vagalumes. Se  $\gamma$  tender ao infinito resultará em uma atratividade aproximadamente nula fazendo com que o vagalume mova-se apenas aleatoriamente. Foram observados valores de  $\beta_0$  e  $\gamma$  variando entre 0 e 1.

Os valores desses parâmetros utilizados tanto no FA-LBG quanto no FA-LBG modificado foram obtidos a partir de uma análise prévia em que foram realizadas 10 execuções em vários conjuntos de parâmetros para cada 10 inicializações diferentes. Foram calculadas as médias de relação sinal-ruído de pico (PSNR, *Peak Signal-to-Noise Ratio*) para cada conjunto de parâmetros e optou-se pelos parâmetros que resultaram nos maiores valores de PSNR.

Seja  $v_p$  o valor de pico de amplitude do sinal de entrada e  $I$  e  $I'$  duas imagens digitais de tamanho  $T_1 \times T_2$  pixels, sendo  $I$  a imagem original e  $I'$  a imagem reconstruída depois de passar pelo processo de quantização vetorial. No caso de uma imagem codificada em 256 níveis de cinza (8 *bpp*), tem-se  $v_p = 255$  e a PSNR definida como

$$PSNR (dB) = 10 \log_{10} \left[ \frac{v_p^2}{MSE} \right], \quad (12)$$

em que MSE (*Mean Square Error*) é o erro médio quadrático entre as imagens  $I$  e  $I'$ , definido como

$$MSE = \frac{1}{T_1 \times T_2} \sum_{l=0}^{T_1-1} \sum_{c=0}^{T_2-1} [I(l, c) - I'(l, c)]^2, \quad (13)$$

em que  $I(l, c)$  e  $I'(l, c)$  representam respectivamente os valores de *pixels* da imagem original e reconstruída na  $l$ -ésima linha e  $c$ -ésima coluna, com  $T_1$  representando o número de *pixels* dispostos horizontalmente e  $T_2$  o número de *pixels* dispostos verticalmente em uma imagem.

Os parâmetros utilizados nos algoritmos foram:  $\alpha = 0,7$ ,  $\beta = 0,4$  e  $\gamma = 0,001$ .

Como critério de parada da versão modificada do FA-LBG e do algoritmo LBG foi utilizado o limiar de distorção  $\epsilon = 0,001$ , enquanto que no algoritmo FA-LBG foi utilizado o número de ciclos igual a 10. Esse valor foi escolhido após analisar o desempenho do FA-LBG, em termos de PSNR, utilizando no projeto 10, 20, 50 e 100 ciclos, observou-se que o PSNR varia pouco com o aumento do número de ciclos e por esse motivo optou-se por utilizar o critério de parada do FA-LBG, 10 ciclos.

Foram executadas 10 inicializações diferentes para cada tamanho  $N$  de dicionário para o projeto com os algoritmos LBG, FA-LBG e FA-LBG modificado. A qualidade dos dicionários projetados foi avaliada por meio dos valores de PSNR obtidos das imagens reconstruídas. Ao final das simulações, os valores de PSNR médio foram computados. O desempenho em termos de PSNR das imagens Boat, Clock e Lena, para cada tamanho de dicionário  $N$  projetado utilizando como conjunto de treino a mesma imagem utilizada como imagem de entrada é apresentado na Tabela I. A Tabela II apresenta o desvio padrão dos valores de PSNR médio apresentados na Tabela I.

Tabela I. VALORES DE PSNR MÉDIO EM *dB*.

Imagem	Método	$N$				
		32	64	128	256	512
Boat	LBG	24,95	25,93	26,91	27,87	29,00
	FA-LBG	24,98	25,92	26,87	27,88	29,04
	FA-LBG modificado	25,01	26,06	27,24	28,60	30,51
Clock	LBG	26,37	27,34	28,31	29,46	30,79
	FA-LBG	26,38	27,43	28,31	29,56	30,77
	FA-LBG modificado	26,69	27,89	29,15	30,76	32,57
Lena	LBG	26,61	27,73	28,82	29,88	31,12
	FA-LBG	26,63	27,71	28,79	29,84	31,13
	FA-LBG modificado	26,70	27,93	29,23	30,78	32,88

Tabela II. DESVIO PADRÃO.

Imagem	Método	$N$				
		32	64	128	256	512
Boat	LBG	0,01	0,02	0,01	0,02	0,22
	FA-LBG	0,03	0,06	0,05	0,07	0,06
	FA-LBG modificado	0,02	0,04	0,02	0,06	0,03
Clock	LBG	0,06	0,09	0,05	0,05	0,03
	FA-LBG	0,19	0,27	0,16	0,15	0,19
	FA-LBG modificado	0,11	0,14	0,11	0,15	0,14
Lena	LBG	0,02	0,02	0,02	0,02	0,04
	FA-LBG	0,04	0,08	0,04	0,10	0,12
	FA-LBG modificado	0,03	0,04	0,03	0,04	0,04

Analisando a Tabela I, os resultados mostram que a versão original do algoritmo FA-LBG para todas as imagens utilizadas apresenta desempenho semelhante ao apresentado pelo algoritmo LBG, produzindo dicionários que levaram a valores similares de PSNR das imagens reconstruídas. Já o algoritmo FA-LBG modificado apresenta desempenho superior ao apresentado tanto pelo algoritmo FA-LBG quanto pelo algoritmo LBG para todos os tamanhos de dicionários para a imagem Clock, e para as demais imagens, o FA-LBG modificado apresenta desempenho levemente superior em relação a sua versão original e ao LBG para os tamanhos de dicionário  $N = 32$  e  $N = 64$  e desempenho superior para os tamanhos  $N = 128, 256$  e  $512$ . Para a imagem Clock a diferença de desempenho dos algoritmos FA-LBG modificado e LBG gira em torno de 0,32 dB a 1,78 dB. Para o tamanho de dicionário 32, o FA-LBG modificado supera o LBG em 0,32 dB, e para o tamanho de dicionário 512, o FA-LBG modificado supera o LBG em 1,78 dB. Para a imagem Boat, essa diferença está compreendida entre 0,06 dB, para  $N = 32$ , e 1,51 dB, para  $N = 512$ , e está compreendida para a imagem Lena entre 0,09 dB, para  $N = 32$ , e 1,76 dB, para  $N = 512$ .

Realizando o teste de Wilcoxon com 5% de significância (95% de confiabilidade) entre o FA-LBG modificado e o FA-LBG e entre o FA-LBG modificado e o LBG, foi observado que o valor de  $p$ -value apresentou-se sempre menor que 0,05, como mostra a Tabela III. Isto significa que as amostras são consideradas diferentes e que a versão modificada do FA-LBG apresenta desempenho superior tanto em relação a sua versão

original quanto em relação ao algoritmo LBG.

Tabela III. TESTE DE WILCOXON PARA OS ALGORITMOS FA-LBG MODIFICADO, FA-LBG E LBG.

Imagem	Método	FA-LBG modificado				
		N				
		32	64	128	256	512
Boat	FA-LBG	3,07e-02	8,78e-05	8,34e-05	8,88e-05	8,88e-05
	LBG	7,92e-05	7,87e-05	8,21e-05	8,68e-05	8,73e-05
Clock	FA-LBG	3,68e-04	1,76e-03	8,98e-05	8,93e-05	8,93e-05
	LBG	8,83e-05	8,78e-05	8,68e-05	8,48e-05	8,93e-05
Lena	FA-LBG	4,31e-04	9,03e-05	8,83e-05	8,83e-05	9,03e-05
	LBG	1,81e-04	8,83e-05	7,92e-05	8,53e-05	8,78e-05

## VII. CONCLUSÃO

Este trabalho propôs modificações na técnica que utiliza computação bio-inspirada para projeto de dicionários denominada FA-LBG, que é baseada no algoritmo do vagalume, aplicada ao projeto de dicionários para QV. As modificações consistiram na forma de inicialização dos algoritmos, na introdução do cálculo dos centróides à etapa de atualização dos vetores-código do dicionário e na definição do critério de parada, resultando no algoritmo FA-LBG modificado.

A técnica foi avaliada por meio dos resultados apresentados em termos de relação sinal-ruído de pico das imagens reconstruídas, que foram comparadas com os resultados obtidos com a versão originalmente apresentada na literatura e com o tradicional algoritmo LBG. Os resultados mostram que, fixados os dicionários iniciais, a versão modificada produz dicionários que levam a maiores valores de PSNR das imagens reconstruídas quando comparados aos dicionários obtidos tanto com a sua versão original quanto com o LBG, para todos os tamanhos de dicionários. Para a imagem Boat e utilizando dicionários de tamanho  $N = 512$ , o FA-LBG modificado conseguiu obter um desempenho superior de 1,51 dB sobre o LBG, de 1,78 dB para a imagem Clock e para a imagem Lena um desempenho superior de 1,76 dB.

Podem-se destacar as seguintes propostas para trabalhos futuros: obter melhoramento do algoritmo proposto a partir do estudo minucioso das mudanças de posição dos vagalumes no espaço de busca; aplicar técnicas de busca eficiente do vizinho mais próximo à etapa de particionamento no projeto de dicionário com o intuito de reduzir a complexidade computacional do projeto e propor famílias de algoritmos FA aplicados ao projeto de dicionário de QV, utilizando para isto outros algoritmos destinados ao projeto de dicionário.

## AGRADECIMENTOS

Os autores agradecem o apoio financeiro do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e da Fundação de Amparo à Ciência e Tecnologia do Estado de Pernambuco (FACEPE).

## REFERÊNCIAS

[1] F. Madeiro e W. T. A. Lopes, "Introdução à Compressão de Sinais," *Revista de Tecnologia da Informação e Comunicação*, vol. 1, n. 1, pp. 33–40, Outubro 2011.

[2] N. Jayant, "Signal Compression: Technology Targets and Research Directions," *IEEE Journal on Selected Areas in Communications*, vol. 10, n. 5, pp. 796–818, June 1992.

[3] N. Jayant, J. Johnston, e R. Safranek, "Signal Compression Based on Models of Human Perception," *Proceedings of the IEEE*, vol. 81, n. 10, pp. 1385–1422, October 1993.

[4] R. M. Gray, "Vector Quantization," *IEEE ASSP Magazine*, pp. 4–29, April 1984.

[5] A. Gersho e R. M. Gray, *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, Boston, MA, 1992.

[6] J.-D. Lee, Y.-H. Chiou, e J.-M. Guo, "Information Hiding Based on Block Match Coding for Vector Quantization-Compressed Images," *IEEE Systems Journal*, vol. 8, n. 3, pp. 737–748, 2014.

[7] J.-D. Lee, Y.-H. Chiou, e J.-M. Guo, "Lossless Data Hiding for VQ Indices Based on Neighboring Correlation," *Information Sciences*, vol. 221, pp. 419–438, February 2013.

[8] C. Qin, C.-C. Chang, e Y.-C. Chen, "A Novel Reversible Data Hiding Scheme for VQ-compressed Images Using Index Set Construction Strategy," *KSII Transactions on Internet and Information Systems*, vol. 7, n. 8, pp. 2027–2041, 2013.

[9] A. Mishra, C. Agarwal, A. Sharma, e P. Bedi, "Optimized Gray-Scale Image Watermarking Using DWT-SVD and Firefly Algorithm," *Expert Systems with Applications*, vol. 41, n. 17, pp. 7858 – 7867, 2014.

[10] Y. Zhou, R. Wang, e D. Yan, "An Audio Watermarking Scheme Based on VQ Codebook Pairing," in *2010 3rd International Congress on Image and Signal Processing*. IEEE, October 2010, pp. 4011–4015.

[11] J.-J. Shen e J.-M. Ren, "A Robust Associative Watermarking Technique Based on Vector Quantization," *Digital Signal Processing*, vol. 20, n. 5, pp. 1408–1423, 2010.

[12] M.-H. Horng e T.-W. Jiang, "Image Vector Quantization Algorithm Via Honey Bee Mating Optimization," *Expert Systems With Applications*, vol. 38, n. 3, pp. 1382–1392, March 2011.

[13] M.-H. Horng, "Vector Quantization Using the Firefly Algorithm for Image Compression," *Expert Systems With Applications*, vol. 39, n. 1, pp. 1078–1091, January 2012.

[14] D. Tsolakis, G. E. Tsekouras, A. D. Niros, e A. Rigos, "On The Systematic Development of Fast Fuzzy Vector Quantization for Grayscale Image Compression," *Neural Networks*, vol. 36, pp. 83–96, September 2012.

[15] C.-W. Tsai, S.-P. Tseng, C.-S. Yang, e M.-C. Chiang, "PREACO: A Fast Ant Colony Optimization for Codebook Generation," *Applied Soft Computing*, vol. 13, n. 6, pp. 3008–3020, January 2013.

[16] Y. Linde, A. Buzo, e R. Gray, "An Algorithm for Vector Quantizer Design," *IEEE Transactions on Communications*, vol. 28, n. 1, pp. 84–95, January 1980.

[17] Q. Chen, J. Yang, e J. Gou, "Image Compression Method Using Improved PSO Vector Quantization," in *Advances in Natural Computation*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2005, vol. 3612, pp. 490–495.

[18] S. Lukasik e S. Zak, "Firefly Algorithm for Continuous Constrained Optimization Tasks," in *Proceedings of the 1st International Conference on Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems*, ser. ICCCI '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 97–106.

[19] S. P. Lloyd, "Least Squares Quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, n. 2, pp. 129–137, March 1982.

[20] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*. Luniver Press, 2008.

[21] I. Fister Jr, X.-S. Yang, e J. Brest, "A Comprehensive Review of Firefly Algorithms," *Swarm and Evolutionary Computation*, vol. 13, pp. 34–46, 2013.

[22] X.-S. Yang, "Firefly Algorithms, Stochastic Test Functions and Design Optimization," *International Journal of Bio-Inspired Computation*, pp. 78–84, 2010.

[23] X.-S. Yang e X. He, "Firefly Algorithms: Recent Advances and Applications," *International Journal of Swarm Intelligence*, pp. 36–50, 2013.

[24] USC, "SIPI Image Database - Misc," 2014. [Online]. Disponível em: <http://sipi.usc.edu/database/database.php>