

Algoritmo PSO Modificado Aplicado ao Projeto de Quantizadores Vetoriais

Verusca Severo*, Herbert A. de Sá Leitão†, Juliano B. Lima‡, 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 de Pernambuco-UPE, Recife, Pernambuco

Email: madeiro@poli.br

Resumo—A quantização vetorial (QV) é amplamente utilizada em compressão de voz e imagem. O desempenho de sistemas de compressão de sinais baseados em QV depende da qualidade dos dicionários projetados. O objetivo do projeto de dicionário é a construção de um dicionário que minimize a distorção introduzida ao se representarem os vetores de entrada por suas correspondentes versões quantizadas. Este trabalho introduz modificações na técnica PSO-LBG (*particle swarm optimization - Linde-Buzo-Gray*), a qual utiliza o algoritmo de otimização por enxame de partículas, PSO, para a construção do dicionário. A versão proposta difere da originalmente apresentada na literatura sobretudo por permitir uma maior influência do conjunto de treino no projeto de dicionários. É apresentada uma avaliação comparativa da técnica com sua versão original e com o tradicional algoritmo LBG (Linde-Buzo-Gray) no que diz respeito à qualidade dos dicionários projetados, avaliada via relação sinal-ruído de pico de imagens reconstruídas.

Palavras-chave: Quantização vetorial, compressão de imagens, projeto de dicionários, otimização por enxame de partículas.

I. INTRODUÇÃO

O crescimento da *Internet*, o avanço tecnológico e a popularização de produtos, como computadores, celulares e *tablets*, levaram ao aumento da quantidade de informação digital transmitida e armazenada, como áudio, imagem, vídeo e texto. Diante do aumento significativo do uso de dados em formato digital, surge a necessidade de utilizar de forma eficaz os recursos disponíveis pelas redes de comunicações. Para isso, faz-se necessário codificar de modo eficiente os sinais para que se adequem às limitações das redes de comunicações. Nesse cenário, a quantização vetorial [1], [2] se destaca como uma poderosa técnica utilizada para compressão de sinais com perdas, em que o sinal quantizado corresponde a uma versão degradada do sinal original com uma quantidade menor de bits. Sua função é realizar o mapeamento de uma sequência de vetores de entrada x que pertencem a um espaço de dimensão K em uma sequência de vetores de mesma dimensão pertencentes a um subconjunto finito desse espaço, denominado dicionário. A quantização vetorial tem sido utilizada com sucesso em sistemas de codificação de sinais, permitindo elevadas taxas de compressão, e tem sido utilizada também em outros sistemas de processamento de sinais, como é o caso de sistemas de esteganografia [3], [4] e marca d'água digital [5], [6].

Entretanto, o desempenho de sistemas de processamento de sinais baseados em quantização vetorial depende da qualidade dos dicionários projetados, visto que a quantização vetorial introduz distorção ao se representarem os vetores de entrada pelos vetores do dicionário que apresentem maior semelhança [1]. Por esse motivo, o estudo de técnicas para geração de dicionários é um tema de pesquisa bastante explorado [7]–[14]. Dentre vários métodos conhecidos para a produção de um dicionário, o algoritmo LBG, proposto por Linde, Buzo e Gray em 1980 [15] se destaca por ser o mais amplamente utilizado. O algoritmo LBG corresponde à generalização do algoritmo de Lloyd [16] para o projeto de quantizadores escalares, que foi estendido para ser aplicado a mais de uma dimensão.

A área de estudos de computação bio-inspirada é responsável pelo desenvolvimento de algoritmos baseados em comportamentos sociais ou em processos naturais [17]–[19]. O uso de técnicas bio-inspiradas como em [7], [9], [13] vem contribuindo para melhorar o processo de quantização vetorial. Em 2005, Chen *et al.* [20] apresentaram um novo método de projeto de dicionários utilizando a técnica de otimização por enxame de partículas (PSO, *Particle Swarm Optimization*). No método proposto, PSO-LBG, cada partícula corresponde a um dicionário que representa uma possível solução ao problema. Uma das partículas corresponde a um dicionário projetado pelo algoritmo LBG, que representará inicialmente a melhor partícula global (*gbest*) do enxame, enquanto as demais partículas correspondem a dicionários iniciais gerados aleatoriamente a partir de um conjunto de treino. As partículas “voam” sobre o espaço vetorial com uma velocidade que é ajustada dinamicamente de acordo com a sua experiência, que corresponde à melhor posição encontrada por ela, e a experiência de seus vizinhos, que corresponde à melhor posição encontrada pelo enxame como um todo. O objetivo do método é encontrar a partícula, ou seja, o dicionário, que apresente menor distorção para um determinado conjunto de vetores.

Este trabalho propõe modificações no algoritmo PSO-LBG com o objetivo de melhorar a qualidade dos dicionários projetados. A versão proposta difere da originalmente apresentada na literatura sobretudo por permitir uma maior influência do conjunto de treino no projeto dos dicionários. As modificações dizem respeito à forma de inicialização, à etapa de treinamento

e à definição do critério de parada do algoritmo.

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 de otimização por enxame de partículas, a Seção IV descreve o funcionamento do PSO-LBG, a Seção V descreve o funcionamento do PSO-LBG modificado, os resultados são apresentados na Seção VI e as conclusões na Seção VII.

II. ALGORITMO LBG

O projeto de dicionários se destaca como um dos problemas mais relevantes da quantização vetorial [1], [2]. O dicionário $W = \{w_i, i = 1, 2, 3, \dots, N\}$ é formado por N vetores-código w_i , ou vetores de reconstrução, com $w_i = \{w_{i1}w_{i2} \dots w_{iK}\}$, em que K é a dimensão do quantizador vetorial e N é o tamanho do dicionário. O erro médio de quantização ao se representar o sinal de entrada por sua versão quantizada é chamado distorção do quantizador. O objetivo do projeto de dicionário é encontrar N vetores-código para um determinado conjunto de treino de forma a reduzir a distorção introduzida ao se representarem os vetores de entrada por suas correspondentes versões quantizadas.

Seja $X = \{x_m, m = 1, 2, 3, \dots, M\}$ o conjunto de treino formado por M vetores treino x_m de dimensão K . Inicialmente, X é particionado em N regiões de Voronoi S_n formando o dicionário inicial W com N vetores-código, $W = \{w_n, n = 1, 2, 3, \dots, N\}$. Cada região de Voronoi é representada por um único vetor-código w_n . Para cada vetor treino x_m , procura-se no dicionário o vetor-código que mais se assemelhe a ele, segundo uma medida de distância, e esse vetor treino é alocado à região de Voronoi correspondente ao vetor-código escolhido, de acordo com

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

Feito o mapeamento dos vetores do conjunto de treino às regiões de Voronoi, calcula-se a distorção somando as distâncias entre cada vetor de treino e o correspondente vetor-código da região à qual foi alocado, de acordo com

$$D_t = \sum_{n=1}^N \sum_{x_m \in S_n} d(x_m, w_n), \quad (2)$$

em que t corresponde a t -ésima iteração.

A partir da distorção, avalia-se o critério de parada do algoritmo LBG, que é encerrado quando

$$\frac{(D_{t-1} - D_t)}{D_t} \leq \epsilon, \quad (3)$$

em que ϵ corresponde ao limiar de distorção.

Caso o critério de parada não seja atingido, os vetores-código do dicionário são calculados e atualizados como os centroides das regiões de Voronoi, segundo o cálculo da média aritmética da j -ésimas componentes de todos os vetores de treino alocados na 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 .

Seja t a iteração do algoritmo LBG que pode ser assim descrito:

- Passo 1: Dados um conjunto de treino X e um dicionário inicial W_0 , faça $t = 0$ e $D_{-1} = \infty$;
- Passo 2: 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: Calcule a distorção total D_t , de acordo com a Equação 2.
- Passo 4: Se o critério de parada definido na Equação 3 for satisfeito, o algoritmo para, com W_t representando o dicionário final, caso contrário continua.
- Passo 5: Calcule e atualize os vetores-código do dicionário de acordo com a Equação 4. Faça $t = t + 1$ e retorne ao *Passo 2*.

III. OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS

Inspirado no comportamento social apresentado pelos pássaros durante a revoada, Kennedy e Eberhart desenvolveram o algoritmo de otimização por enxames de partículas (PSO, *Particle Swarm Optimization*) [21]. Eles se basearam na interação entre as aves ao longo do vôo na busca por comida ou de um lugar para o ninho e modelaram essas características de comportamento como um mecanismo de otimização.

No algoritmo PSO, o enxame corresponde ao bando de pássaros, o espaço de busca corresponde à área sobrevoada pelos pássaros e cada partícula simboliza um pássaro, que corresponde a um ponto mapeado no espaço de busca e representa uma possível solução do problema a ser resolvido. Cada partícula possui um vetor posição que representa a localização de cada pássaro durante o vôo. A solução ótima simboliza a localização do ninho ou alimento.

Uma função *fitness* é utilizada para avaliar o desempenho de cada partícula. A partir dela define-se o *pbest* e o *gbest*. As partículas deslocam-se no espaço de busca com base em sua experiência individual e na experiência coletiva do bando [22]. A experiência individual de cada partícula é representada pela posição *personal best*, o \overrightarrow{pbest} , que define a melhor posição conhecida pela partícula. A experiência coletiva é representada pela posição *global best*, o \overrightarrow{gbest} , que define a melhor posição conhecida pelo enxame como um todo. Seja \vec{p} a posição de cada partícula. A função que descreve a atualização do vetor velocidade, no tempo $t + 1$, é definida por [23]

$$\overrightarrow{v}_{ik}^{t+1} = \overrightarrow{v}_{ik}^t + c_1 r_1^t (\overrightarrow{pbest}_{ik}^t - \overrightarrow{p}_{ik}^t) + c_2 r_2^t (\overrightarrow{gbest}_k^t - \overrightarrow{p}_{ik}^t), \quad (5)$$

em que k representa a dimensão do espaço de busca, i é a i -ésima partícula do enxame, c_1 é a constante de aceleração cognitiva da partícula utilizada para controlar a atração da partícula em direção à melhor posição encontrada por ela

mesmo, c_2 é a constante de aceleração social utilizada para controlar a atração da partícula em direção à melhor posição encontrada pelo bando e r_1 e r_2 são coeficientes aleatórios que variam de 0 a 1.

A partir do vetor velocidade atualizado, calcula-se a nova posição da partícula, conforme

$$\vec{p}_{ik}^{t+1} = \vec{p}_{ik}^t + \vec{v}_{ik}^{t+1}. \quad (6)$$

A Figura 1 ilustra o fluxograma do algoritmo PSO. Dada uma população inicial de P partículas p_i , $i = 1, 2, 3, \dots, P$, em que cada partícula representa um candidato à solução, o algoritmo inicia com o cálculo do *fitness* de cada partícula. No caso de um problema de maximização, se a função *fitness* da partícula p_i em determinada iteração for maior que na iteração anterior, o \vec{pbest} é atualizado. Define-se o melhor de todos os \vec{pbest} em cada iteração, se esse valor for maior que \vec{gbest} encontrado até o momento, o \vec{gbest} é atualizado. Com o \vec{pbest} e o \vec{gbest} verificados e atualizados caso seja necessário, as velocidades e as posições de cada partículas são atualizadas. O processo se repete até que o critério de parada seja atingido.

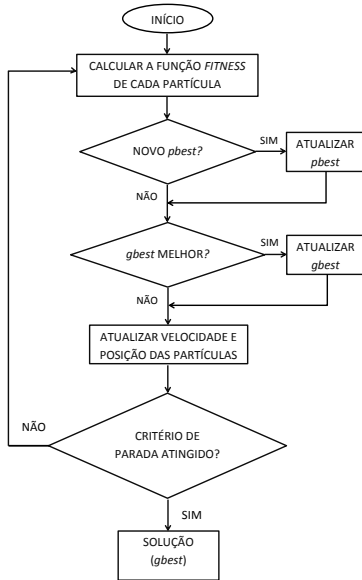


Figura 1. Fluxograma do Algoritmo PSO.

IV. PSO-LBG

O algoritmo PSO-LBG foi proposto por Chen *et al.* em 2005 e apresenta a aplicação do algoritmo de otimização por enxame de partículas ao projeto de dicionários de quantização vetorial [20]. O algoritmo utiliza como função objetivo a distorção (quanto menor a distorção, maior o *fitness*). O objetivo é obter um dicionário que minimize a distorção.

No PSO-LBG cada partícula, inicialmente, corresponde a um dicionário inicial de tamanho N e dimensão K . Para a inicialização da população o resultado do algoritmo LBG é utilizado como uma das partículas iniciais, enquanto o restante da população é inicializado de forma aleatória a partir de um conjunto treino. Cada partícula tem associada a ela um vetor

posição, que armazena a posição da partícula no espaço de busca e o vetor velocidade, que é responsável por indicar a direção em que ocorrem as mudanças de posição de cada partícula. O vetor posição de cada partícula corresponde ao vetor-código de cada dicionário. Inicialmente, o vetor velocidade é definido como um número aleatório compreendido na faixa de $[-1, 1]$.

Seja p_o a o -ésima partícula do enxame com $1 \leq o \leq P$, que representa o o -ésimo dicionário $W^{(o)} = \{w_i^{(o)}, i = 1, 2, 3, \dots, N\}$, em que $w_i^{(o)}$ corresponde ao i -ésimo vetor-código do dicionário $W^{(o)}$. Com o conjunto de treino particionado em M vetores, a cada iteração o PSO-LBG realiza, para cada partícula, o mapeamento dos M vetores treino nas N regiões de Voronoi e calcula a distorção média para obter o *fitness*, conforme

$$f(p_o) = \frac{1}{D(p_o)}. \quad (7)$$

A partir do valor *fitness* deve-se atualizar a posição \vec{pbest} de cada partícula e a posição \vec{gbest} . Seja \vec{pbest}_{max}^t o maior valor de \vec{pbest} obtido entre todas as partículas na iteração t . A cada iteração do algoritmo, o \vec{pbest} de $W^{(o)}$ e o \vec{gbest} são atualizados da seguinte forma

$$\vec{pbest}^t = \begin{cases} W^{(o)t}, & \text{se } f(W^{(o)t}) > f(W^{(o)t-1}, \\ \vec{pbest}^{t-1}, & \text{caso contrário.} \end{cases} \quad (8)$$

$$\vec{gbest}^t = \begin{cases} \vec{pbest}_{max}^t, & \text{se } \vec{pbest}_{max}^t > \vec{gbest}^{t-1} \\ \vec{gbest}^{t-1}, & \text{caso contrário.} \end{cases} \quad (9)$$

O critério de parada, que é definido pelo número de ciclos do algoritmo, é analisado. Caso seja atingido, o algoritmo para, com a posição \vec{gbest} representando o dicionário final, caso contrário, os vetores velocidade e posição são calculados e atualizados.

O vetor velocidade é atualizado conforme a Equação 5 e a atualização do vetor posição é realizada segundo a Equação 6. Com a atualização do posicionamento das partículas, inicia-se uma nova iteração e o processo se repete até que o critério de parada seja estabelecido.

O algoritmo pode ser resumido na seguinte sequência de passos:

- Passo 1: Inicialize a população com P partículas, sendo uma partícula resultado do algoritmo LBG e as $P - 1$ partículas geradas aleatoriamente a partir de um conjunto de treino, e defina os parâmetros cognitivo c_1 e social c_2 e o número de iterações;
- Passo 2: Particione o conjunto de treino em M vetores de dimensão K ;
- Passo 3: Para cada partícula, faça:
 - aloque os M vetores treino nas N regiões de Voronoi segundo o VMP;
 - calcule a distorção de acordo com a Equação (2);

calcule o *fitness*;
 atualize a posição *pbest*;

Passo 4: Determine a partícula de maior *fitness* e atualize a posição *gbest*;

Passo 5: Se o número de iterações não for alcançado, faça para cada partícula:
 atualize o vetor velocidade conforme as Equações (10) e (5);
 atualize o vetor posição conforme a Equação (6) e retorne ao Passo 3.

V. PSO-LBG MODIFICADO

O algoritmo PSO-LBG modificado foi obtido a partir de alterações realizadas no PSO-LBG, visando promover uma maior influência do conjunto de treino no projeto de dicionários. As alterações dizem respeito à forma de inicialização do algoritmo, à etapa de treinamento do dicionários, ou seja, à etapa de atualização dos vetores-código, e à definição do critério de parada do algoritmo.

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

ETAPA DE INICIALIZAÇÃO: Todas as partículas da população são inicializadas de maneira aleatória a partir de um conjunto treino.

ETAPA DE TREINAMENTO: O reposicionamento das partículas é precedido pelo cálculo dos centroides de cada região de Voronoi. Como o cálculo do centroide de cada região de Voronoi é feito como a média aritmética das componentes de todos os vetores de treino pertencentes à região de Voronoi em questão, o objetivo de introduzir o cálculo dos centroides na etapa de atualização do dicionário no PSO-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 correspondentes. A atualização de cada partícula no PSO-LBG modificado acontece em duas etapas:

- Atualização dos centroides das regiões de Voronoi de cada uma das P partículas, segundo a Equação 4;
- Atualização dos vetores velocidade e posição de cada uma das partículas, segundo as Equações 10, 5 e 6.

No PSO-LBG modificado, o vetor velocidade de cada partícula é atualizado em duas etapas:

- Primeiro calcula-se a velocidade atual v_i^t para cada vetor posição de acordo com a Equação 10, com o objetivo de garantir a convergência do PSO-LBG modificado, de maneira que a partícula siga uma trajetória de convergência baseada nas características do projeto de dicionários do LBG, visto que a atualização do posicionamento das partículas é precedida pelo cálculo dos centroides. Na verdade, a Equação 10 corresponde à diferença entre a atualização da partícula após e antes o cálculo dos centroides.

$$\vec{v}_i^t = \vec{p}_i^t - \vec{p}_i^{t-1}. \quad (10)$$

- Na segunda etapa, a velocidade é atualizada conforme a Equação 5.

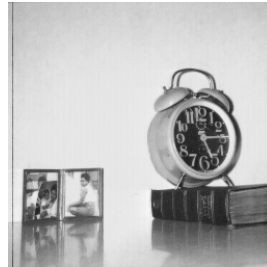
CRITÉRIO DE PARADA: O critério de parada delinea a convergência do algoritmo. A quantização vetorial introduz distorção ao se representarem os vetores de treino pelos seus vizinhos mais próximos. No algoritmo LBG, essa distorção introduzida é monitorada a cada iteração e serve de base para o critério de parada do algoritmo. O algoritmo PSO-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 estiver abaixo de um determinado limiar de distorção.

VI. RESULTADOS

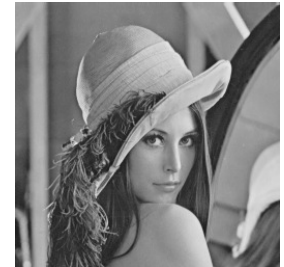
As imagens Boat, Clock e Lena de dimensão 256×256 , no formato *Portable Gray Map* (PGM), codificadas a 8 bits por pixels (*bpp*) (256 níveis de cinza), foram utilizadas como conjunto de treino para o projeto de dicionários e também como sinal de entrada no processo de quantização vetorial. As imagens foram obtidas da base de imagens do *Signal and Image Processing Institute* (SIPI) da *University of Southern California* (USC) [24] e são apresentadas na Figura 2.



(a) Boat.



(b) Clock.



(c) Lena.

Figura 2. Imagens utilizadas nas simulações.

Os dicionários iniciais utilizados nas simulações foram gerados aleatoriamente a partir das imagens de treino. Esse processo de geração de dicionários iniciais se faz com a escolha aleatória de blocos de *pixels* da imagem treino. Os dicionários foram projetados com dimensão (K) igual a 16 (blocos 4×4 *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*.

No PSO-LBG, foram utilizadas $P = 10$ partículas, sendo que uma das partículas corresponde a um dicionário projetado pelo algoritmo LBG e as $(P-1)$ partículas restantes correspondem a dicionários gerados aleatoriamente a partir de uma imagem treino. A versão modificada do PSO-LBG, por sua vez, utiliza como ponto de partida $P = 10$ partículas,

todas geradas aleatoriamente. Para sua inicialização foram utilizados os mesmos ($P-I$) dicionários aleatórios utilizados na versão original do PSO-LBG e mais um dicionário gerado aleatoriamente a partir de uma imagem treino, compondo dessa forma P dicionários iniciais aleatórios.

O critério de parada estabelecido para o projeto utilizando o PSO-LBG foi o número de ciclos igual a 10. Esse valor foi escolhido após analisar o desempenho do PSO-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. Na versão modificada do PSO-LBG e no algoritmo LBG foi utilizado como critério de parada o limiar de distorção $\epsilon = 0,001$.

Os valores dos parâmetros utilizados no PSO-LBG e no PSO-LBG modificado estão apresentados na Tabela I. Esses valores 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.

Tabela I. PARÂMETROS DO PSO-LBG E PSO-LBG MODIFICADO

Imagem	Constante de Aceleração Cognitiva	Constante de Aceleração Social
Boat	$c_1 = 0,9$	$c_2 = 0,7$
Clock	$c_1 = 0,8$	$c_2 = 0,2$
Lena	$c_1 = 0,9$	$c_2 = 0,7$

O PSNR de uma determinada imagem reconstruída é definido como

$$PSNR (dB) = 10 \log_{10} \left[\frac{e^2}{MSE} \right], \quad (11)$$

em que e é o valor de pico de amplitude do sinal de entrada, que no caso de uma imagem codificada em 256 níveis de cinza (8 *bpp*) tem-se $e = 255$, e o MSE (*Mean Square Error*) é o erro médio quadrático entre os *pixels* da imagen original e os *pixels* da imagem reconstruída e é definido por

$$MSE = \frac{1}{E_1 \times E_2} \sum_{l=0}^{E_1-1} \sum_{c=0}^{E_2-1} [H(l, c) - H'(l, c)]^2, \quad (12)$$

em que H corresponde à imagem original e H' à imagem reconstruída depois de passar pelo processo de quantização vetorial, E_1 representa os *pixels* dispostos horizontalmente e E_2 os *pixels* dispostos verticalmente em uma imagem.

Dadas uma imagem original codificada em 8 *bpp* e sua versão quantizada a partir de um processo de quantização vetorial utilizando um dicionário de tamanho N e dimensão K , quanto menor o valor de MSE entre as imagens original e quantizada, maior será o valor de PSNR da imagem quantizada, consequentemente, melhor será a sua qualidade visual.

Foram utilizadas 10 inicializações diferentes na execução dos algoritmos LBG, PSO-LBG e PSO-LBG modificado para cada tamanho N de dicionário. A qualidade dos dicionários gerados foi avaliada por meio dos valores de PSNR médio computados ao final das simulações.

A Tabela II apresenta o desempenho em termos de PSNR para diversos tamanhos de dicionário N para as imagens Boat, Clock e Lena. A Tabela III corresponde ao desvio padrão dos PSNR médio apresentados na Tabela II. Os resultados mostram que a versão original do algoritmo PSO-LBG apresenta desempenho semelhante ao apresentado pelo algoritmo LBG, produzindo dicionários que levaram a valores similares de PSNR das imagens reconstruídas.

Tabela II. 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
	PSO-LBG	24,98	25,92	26,87	27,88	29,04
	PSO-LBG modificado	25,00	26,06	27,22	28,63	30,49
Clock	LBG	26,37	27,34	28,31	29,46	30,79
	PSO-LBG	26,38	27,43	28,31	29,57	30,77
	PSO-LBG modificado	26,78	28,01	29,11	30,40	32,64
Lena	LBG	26,61	27,73	28,82	29,88	31,12
	PSO-LBG	26,63	27,71	28,79	29,84	31,13
	PSO-LBG modificado	26,69	27,93	29,21	30,75	32,79

Tabela III. DESVIO PADRÃO.

Imagem	Método	N				
		32	64	128	256	512
Boat	LBG	0,01	0,02	0,01	0,02	0,22
	PSO-LBG	0,03	0,06	0,05	0,07	0,06
	PSO-LBG modificado	0,01	0,02	0,03	0,04	0,06
Clock	LBG	0,06	0,09	0,05	0,05	0,03
	PSO-LBG	0,19	0,27	0,16	0,15	0,19
	PSO-LBG modificado	0,09	0,10	0,13	0,16	0,48
Lena	LBG	0,02	0,02	0,02	0,02	0,04
	PSO-LBG	0,03	0,08	0,04	0,10	0,12
	PSO-LBG modificado	0,05	0,04	0,04	0,06	0,08

A versão modificada do PSO-LBG apresenta desempenho superior em relação a sua versão original e ao algoritmo LBG para todos os tamanhos de dicionário para a imagem Clock, como mostra a Tabela II. A diferença dos valores de PSNR médio obtidos com a utilização do algoritmo PSO-LBG modificado em relação ao algoritmo LBG está compreendida entre 0,41 dB a 1,85 dB. Por exemplo, para o tamanho de dicionário 32, o algoritmo PSO-LBG modificado supera o algoritmo LBG em 0,41 dB. Já para o tamanho de dicionário 512, o algoritmo PSO-LBG modificado supera o algoritmo LBG em 1,85 dB.

Para as imagens Boat e Lena, o PSO-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 . A diferença dos valores de PSNR médio obtidos entre o algoritmo PSO-LBG modificado e o algoritmo LBG para a imagem Boat está compreendido entre 0,05 dB (para $N = 32$) a 1,49 dB (para $N = 512$). E para a imagem Lena, está compreendido entre 0,08 dB (para $N = 32$) a 1,67 dB (para $N = 512$).

Para avaliar os resultados de PSNR das imagens reconstruídas a partir dos dicionários projetados, foi utilizado o teste de Wilcoxon. A Tabela IV apresenta o teste de Wilcoxon com 5% de significância, ou seja, o teste com 95% de confiabilidade, realizado entre os algoritmos PSO-LBG modificado e o PSO-LBG, e entre o PSO-LBG modificado e o LBG. Os resultados mostram que o valores de p - *value* computados são menores que 0,05. A partir destes resultados, pode-se afirmar que os valores de PSNR médios obtidos com os

algoritmos PSO-LBG modificado e os algoritmos PSO-LBG e LBG são distintos e, então podemos concluir que os algoritmos apresentam desempenho distintos e a versão modificada do PSO-LBG apresenta desempenho superior tanto em relação a sua versão original quanto em relação ao algoritmo LBG.

Tabela IV. TESTE DE WILCOXON PARA OS ALGORITMOS PSO-LBG MODIFICADO, PSO-LBG E LBG.

Imagem	Método	PSO-LBG modificado				
		N				
		32	64	128	256	512
Boat	PSO-LBG	8,20e-02	8,49e-05	8,58e-05	8,93e-05	9,03e-05
	LBG	7,47e-05	7,60e-05	8,44e-05	8,68e-05	8,83e-05
Clock	PSO-LBG	8,83e-05	1,60e-04	8,98e-05	8,98e-05	8,98e-05
	LBG	8,83e-05	8,83e-05	8,68e-05	8,54e-05	8,98e-05
Lena	PSO-LBG	4,46e-03	8,78e-05	8,78e-05	5,41e-06	9,03e-05
	LBG	1,04e-03	8,58e-05	7,87e-05	8,83e-05	9,03e-05

VII. CONCLUSÃO

Este trabalho introduziu modificações no algoritmo PSO-LBG, que utiliza o algoritmo de otimização por enxame de partículas, PSO, para a construção do dicionário de quantização vetorial. Modificações foram introduzidas no algoritmo PSO-LBG, resultando no algoritmo PSO-LBG modificado. As modificações consistiram na forma de inicialização dos algoritmos, na introdução do cálculo dos centroides à etapa de atualização dos vetores-código do dicionário e na definição do critério de parada.

Os algoritmos foram avaliadas por meio dos resultados de PSNR das imagens reconstruídas. Os resultados mostram que as modificações introduzidas no algoritmo produzem dicionários que levam a maiores valores de PSNR das imagens reconstruídas quando comparados aos dicionários obtidos com as suas versões originais. Para a imagem Boat e utilizando dicionários de tamanho $N = 512$, o PSO-LBG modificado conseguiu obter um ganho de 1,49 dB sobre o LBG, de 1,67 dB para a imagem Lena e para a imagem Clock um ganho de 1,85 dB em termos de PSNR.

Como trabalhos futuros pretende-se realizar uma análise “microscópica” do comportamento das partículas, vistoriando o seu PSNR em cada iteração e observando as mudanças de posição no espaço de busca com o intuito de propor melhoramento nos algoritmos; propor famílias de algoritmos PSO aplicados ao projeto de dicionário de quantização vetorial, trocando o algoritmo LBG por outros algoritmos destinados ao projeto de dicionário; propor métodos para acelerar o processo de convergência do PSO-LBG modificado, com a finalidade de reduzir o tempo para o projeto de dicionário, e estudar a aplicação de outros algoritmos bioinspirados ao projeto de dicionários para quantização vetorial.

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

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

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] X. Yan, Y. Zhu, W. Zou, e L. Wang, “A New Approach for Data Clustering Using Hybrid Artificial Bee Colony Algorithm,” *Neurocomputing*, vol. 97, pp. 241 – 250, 2012.

[9] 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.

[10] 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.

[11] A. Hatamlou, S. Abdullah, e H. Nezamabadi-pour, “A Combined Approach for Clustering Based on K-means and Gravitational Search Algorithms,” *Swarm and Evolutionary Computation*, vol. 6, pp. 47 – 52, 2012.

[12] Y.-C. Hu, W.-L. Chen, C.-C. Lo, C.-M. Wu, e C.-H. Wen, “Efficient VQ-Based Image Coding Scheme Using Inverse Function and Lossless Index Coding,” *Signal Processing*, vol. 93, n. 9, pp. 2432–2439, September 2013.

[13] 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.

[14] S. J. Nanda e G. Panda, “A Survey on Nature Inspired Metaheuristic Algorithms for Partitional Clustering,” *Swarm and Evolutionary Computation*, vol. 16, pp. 1 – 18, 2014.

[15] 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.

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

[17] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*, 2ª ed. University of Cambridge, United Kingdom: Luniver Press, 2010.

[18] L. N. de Castro e F. J. Von Zuben, *Recent Developments in Biologically Inspired Computing*. IGI Global, 2005.

[19] L. N. de Castro, “Fundamentals of Natural Computing: An Overview,” *Physics of Life Reviews*, vol. 4, n. 1, pp. 1–36, 2007.

[20] 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.

[21] J. Kennedy e R. Eberhart, “Particle Swarm Optimization,” *Proceedings of IEEE International Conference on Neural Networks*, pp. 1942–1948, 1995.

[22] D. Floreano e C. Mattiussi, *Bio-inspired Artificial Intelligence: Theories, Methods and Technologies*. MIT Press, Cambridge, MA, 2008.

[23] X.-S. Yang, Z. Cui, R. Xiao, A. H. Gandomi, e M. Karamanoglu, *Swarm Intelligence and Bio-inspired Computation: Theory and Applications*, 1ª ed. Elsevier, 2013.

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