

Técnicas de Busca do Vizinho Mais Próximo Aplicadas ao Projeto de Dicionários PSO-COVQ

Felipe Alberto B. S. Ferreira
Universidade de
Pernambuco (UPE)
Pernambuco, Brasil
Email: fabsf@ecomp.poli.br

Herbert A. de Sá Leitão
Universidade Federal
de Pernambuco (UFPE)
Pernambuco, Brasil
Email: herbert.leitao@ufpe.br

Waslon T. A. Lopes
Universidade Federal
da Paraíba (UFPB)
Paraíba, Brasil
Email: waslon@ieee.org

F. Madeiro
Universidade de
Pernambuco (UPE)
Pernambuco, Brasil
Email: madeiro@poli.br

Resumo—Um problema desafiador relacionado à quantização vetorial (QV) é o projeto de dicionários com robustez a erros de canal. Nesse cenário, vem sendo usada a quantização vetorial otimizada para canal (COVQ, Channel-Optimized Vector Quantization). O projeto de dicionários COVQ pode ser realizado pela otimização por enxame de partículas (PSO, Particle Swarm Optimization). Neste artigo, técnicas eficientes de busca pelo vizinho mais próximo (VMP) são acomodadas no PSO para COVQ, levando a uma redução de tempo de até 94,3% quando comparadas ao PSO convencional para COVQ de imagens.

Keywords—busca pelo vizinho mais próximo, quantização vetorial de imagens, COVQ, PSO.

I. INTRODUÇÃO

A compressão de sinais desempenha um papel crucial em aplicações que envolvem transmissão e/ou armazenamento de sinais. O alvo das técnicas de compressão de sinais é reduzir o número de bits utilizados na representação dos sinais, dentre os quais podem ser citados imagem, vídeo, áudio e voz. A quantização vetorial (QV) [1], [2] tem sido utilizada em diversos estudos de compressão de sinais e apresenta, segundo a Teoria da Distorção *versus* Taxa, uma superioridade sobre a quantização escalar. Muitos trabalhos utilizam esta técnica em diversas aplicações como, por exemplo, na compressão de imagens [3], [4], em projetos de alocação de antenas para telefonia celular [5], em processamento de imagens médicas [6], em esteganografia [7] e marca d'água digital [8].

Um sistema eficiente de compressão de sinais reduz, na medida do possível, as partes redundante e irrelevante da informação, conservando a parcela interessante, ou seja, a parcela imprescindível para o receptor [9], como um esforço para aumentar a taxa de compressão. Em contrapartida, a remoção da redundância pode introduzir, no projeto de dicionários, um alto nível de sensibilidade aos ruídos do canal de transmissão [10]. Especificamente para a quantização vetorial com imagens, esses ruídos geram bloqueamentos espúrios comprometendo a qualidade final da imagem reconstruída quando comparada à imagem original [11]–[13]. Uma abordagem para minimizar este impacto é utilizar um quantizador vetorial otimizado para canal (COVQ, Channel-Optimized Vector Quantization) [10], [14], que tem como premissa usar características do próprio canal de transmissão no projeto de dicionário da QV. Desta forma é possível obter uma melhoria na qualidade do sinal reconstruído [10]. Uma alternativa é utilizar técnicas bio-inspiradas para reduzir a distorção média final

dos dicionários produzidos, como a aplicação da otimização por enxame de partículas (PSO, Particle Swarm Optimization) [15] ao projeto de dicionários do COVQ, método denominado PSO-COVQ [11]. No entanto, o algoritmo PSO tem complexidade computacional maior que a do algoritmo clássico de projeto de dicionário COVQ. Além disso, a complexidade computacional do PSO-COVQ é proporcional à quantidade de partículas utilizadas. O presente trabalho apresenta, no cenário de PSO-COVQ, a aplicação das técnicas *Partial Distance Search* (PDS) e *Equal-average Equal-variance Equal-norm Nearest Neighbor Search* (EEENNS). Trata-se de um novo cenário de aplicação das técnicas supracitadas, originalmente utilizadas na etapa de codificação por distância mínima.

O restante deste artigo encontra-se organizado da seguinte forma: a Seção II aborda a quantização vetorial otimizada para canal, a Seção III apresenta brevemente a meta-heurística de otimização por enxames de partículas e o funcionamento do PSO-COVQ, a Seção IV descreve as técnicas de aceleração utilizadas, os resultados são discutidos na Seção V e a conclusão apresentada na Seção VI.

II. QUANTIZAÇÃO VETORIAL OTIMIZADA PARA CANAL

A QV é um processo de mapeamento

$$Q: \mathbb{R}^K \rightarrow W, \quad (1)$$

de um espaço euclidiano K -dimensional em um subconjunto finito deste espaço. O subconjunto é denominado de dicionário $W = \{\vec{W}_i, i = 1, 2, \dots, N\}$, $\vec{W}_i \in \mathbb{R}^K$, em que \vec{W}_i é o centróide ou vetor-código da i -ésima região de Voronoi S_i . O mapeamento de um vetor de entrada \vec{X} em um vetor-código é realizado utilizando o critério de distância mínima (vizinho mais próximo, VMP). No estudo foi adotada a distância euclidiana quadrática, definida por

$$d(\vec{X}, \vec{W}_i) = \sum_{j=1}^K (X_j - W_{ij})^2, \quad (2)$$

em que X_j é a j -ésima componente do vetor de entrada \vec{X} e W_{ij} é a j -ésima componente do i -ésimo vetor-código.

Seja $\xi = \{i, i = 1, 2, \dots, N\}$ um conjunto de indexação que representa os símbolos transmitidos pelo canal. Tem-se a relação

$$W \times \xi = \{(\vec{W}_i, i) | \vec{W}_i \in W \wedge i \in \xi\}. \quad (3)$$

A codificação do vetor de treino em uma dada região de Voronoi S_i é dada por $\xi(\vec{X}) = i \leftrightarrow VMP(\vec{X}) = \vec{W}_i$.

Seja $P_{y|x}(j|i)$ a probabilidade de transmitir i ($i \in \xi$) e receber j ($j \in \xi$), devido ao erro do canal. Para um modelo de canal binário simétrico, tem-se que a distorção média é definida como

$$D = \frac{1}{kM} \sum_{m=1}^M \sum_{j=1}^N P_{y|x}(j|\xi(\vec{X}_m)) \times d(\vec{X}_m, \vec{W}_j). \quad (4)$$

O centróide da j -ésima região de Voronoi é dado por

$$\vec{W}_j = \frac{\sum_{i=1}^N P_{y|x}(j|i)}{\sum_{i=1}^N P_{y|x}(j|i)|S_i|} \sum_{S_i} \vec{X}_m, \quad (5)$$

em que $|S_i|$ é o número de vetores de treino pertencentes à região de Voronoi S_i .

III. OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS

A otimização por enxame de partículas [15] é uma técnica de otimização inspirada no comportamento social de pássaros durante as revoadas. O processo é inicializado com uma população de soluções aleatórias e procura por uma solução ótima com o passar das iterações. No PSO, as soluções em potencial, chamadas de partículas, “voam” pelo “espaço” do problema seguindo a atual partícula “ótima”.

As soluções local (melhor posição já encontrada por uma determinada partícula) e global (melhor posição já encontrada pelo enxame como um todo) são chamadas de posição *pbest* (*personal best*) e *gbest* (*global best*), respectivamente. Para definir estas posições é utilizada uma função denominada *fitness*, definida como o inverso da distorção média no caso de projeto de dicionário para COVQ. Esta função é usada para avaliar o grau de aptidão das partículas. Inicialmente, a posição *pbest* de cada partícula é representada por sua posição inicial e, caso sua aptidão seja melhor nas iterações seguintes, sua posição é atualizada. Já a posição *gbest* é representada inicialmente pela posição da partícula que tem maior *fitness*. Nas iterações seguintes esta posição somente é atualizada caso alguma partícula obtenha uma aptidão melhor.

A. PSO aplicado à COVQ

No caso do PSO aplicado ao projeto de dicionários de COVQ [11], cada partícula é representada por um dicionário diferente e movimenta-se no espaço utilizando um vetor de velocidade \vec{v} , que é atualizado da seguinte forma. Primeiro, calcula-se o valor inicial para cada vetor-código conforme a Equação (6), de forma que a partícula siga uma trajetória de convergência baseada no projeto de dicionário COVQ.

$$\vec{v}_{lj[n]} = \vec{W}_{lj[n]} - \vec{W}_{lj[n-1]}, \quad (6)$$

em que $\vec{W}_{lj[n]}$ é o centróide j da partícula l na n -ésima iteração e $\vec{W}_{lj[n-1]}$ é o centróide j da partícula l na iteração anterior a iteração n . Em seguida a velocidade de cada vetor-código é atualizada conforme a equação

$$\vec{v}_{lj} \leftarrow \vec{v}_{lj} + c_1 r_1 (\overline{pbest}_{lj} - \vec{W}_{lj}) + c_2 r_2 (\overline{gbest}_j - \vec{W}_{lj}), \quad (7)$$

em que c_1 é o parâmetro cognitivo da partícula, c_2 é o parâmetro de aprendizado social e r_1 e r_2 são valores aleatórios compreendidos entre 0 e 1.

B. Algoritmo PSO-COVQ

Seguem os passos do algoritmo PSO aplicado ao projeto de dicionários de um quantizador vetorial otimizado para canal:

- *Passo 1:* Inicialmente são gerados aleatoriamente φ_p dicionários iniciais de tamanho N e dimensão K . No PSO-COVQ, cada partícula é representada por um dos dicionários iniciais;
- *Passo 2:* Para cada partícula deve ser realizado o mapeamento dos M vetores de treino nas N regiões de Voronoi de acordo com a distância mínima;
- *Passo 3:* Calcula-se, para cada partícula, o centróide de cada região;
- *Passo 4:* Calcula-se, para cada partícula, a distorção média da iteração atual D_l , em que a partícula é representada por l ;
- *Passo 5:* Para cada partícula, é calculada a aptidão

$$fitness(\varphi_{l[n]}) = \frac{1}{D_l}; \quad (8)$$

- *Passo 6:* Atualiza-se a posição *pbest* e *gbest* de cada partícula;
- *Passo 7:* Aplica-se o teste de convergência na partícula na posição *gbest* (partícula de maior aptidão até o momento). Caso o teste de convergência tenha sucesso o algoritmo é encerrado. A condição de parada do algoritmo depende do limiar de distorção μ estabelecido. O teste de convergência consiste em verificar se $(D_{n-1} - D_n)/D_n \leq \mu$, em que n representa a iteração atual e $n - 1$ a iteração anterior. Desta forma, caso a diferença entre a distorção média da iteração atual e da iteração anterior for menor ou igual ao limiar μ , então pode ser dito que o dicionário possui um erro aceitável em relação ao conjunto de treino e o algoritmo pode ser encerrado. Caso contrário, continua no *Passo 8*;
- *Passo 8:* Atualiza-se o vetor de velocidade de cada vetor-código para cada partícula;
- *Passo 9:* Atualiza-se o vetor posição de cada partícula conforme Equação (9) e repete a partir do *Passo 2*.

$$\vec{W}_{lj} \leftarrow \vec{W}_{lj} - \vec{v}_{lj} \quad (9)$$

IV. TÉCNICAS DE ACELERAÇÃO DA ETAPA DE PARTICIONAMENTO

Esta seção apresenta algumas técnicas de aceleração da etapa de codificação a serem utilizadas na etapa de particionamento do PSO-COVQ. São elas o *Partial Distance Search* (PDS) [16] e os algoritmos *Equal-average Nearest Neighbor Search* (ENNS) [17] e *Equal-average Equal-variance Nearest Neighbor Search* (EENNS) [18] que são base para o algoritmo *Equal-average Equal-variance Equal-norm Nearest Neighbor Search* (EEENNS) [19].

```

1:  $d_{min} \leftarrow \infty$ 
2:  $i \leftarrow 1$ 
3: for  $i = 1$  to  $N$  do
4:    $d \leftarrow 0$ 
5:    $j \leftarrow 1$ 
6:   for  $j = 1$  to  $K$  do
7:      $d \leftarrow d + (X_j - W_{ij})^2$ 
8:     if  $d > d_{min}$  then
9:       interrompe
10:    end if
11:  end for
12:  if  $d < d_{min}$  then
13:     $d_{min} \leftarrow d$ 
14:     $vmp \leftarrow i$ 
15:  end if
16: end for

```

Figura 1: Passo a passo do algoritmo PDS.

A. PDS

O algoritmo de busca por distância parcial (PDS) [16] constitui uma das técnicas tradicionais para redução da complexidade computacional envolvida na etapa de codificação por distância mínima da QV ou etapa de busca pelo vizinho mais próximo. Nesse algoritmo, o codificador toma a decisão antes de completar o cálculo de distância entre o vetor de entrada e um vetor-código.

Na etapa de codificação do projeto de dicionários, decide-se que o vetor-código em questão não representa o vizinho mais próximo se, para algum $j < K$, a distância acumulada (ou seja, a distância parcial) para as primeiras j componentes do vetor de entrada é maior que a menor distância até então encontrada no processo de busca. Na etapa de codificação, então, é interrompido o cálculo de distância para esse vetor-código e inicia esse cálculo para o próximo vetor-código do dicionário. Com essa abordagem, o número de multiplicações por amostra é reduzido. Observa-se, também, redução do número de subtrações/adições por amostra. Muito embora o algoritmo PDS introduza um aumento do número de comparações, a complexidade global observada na fase de procura do vizinho mais próximo é reduzida.

A estrutura do algoritmo PDS é apresentada na Fig. 1, em que d denota a distância (distorção), d_{min} representa a distância mínima (ou seja, a menor distância até então encontrada no processo de busca), X_j denota a j -ésima componente do vetor de entrada K -dimensional \vec{X} e W_{ij} representa a j -ésima componente do i -ésimo vetor-código K -dimensional \vec{W}_i , $i = 1, 2, \dots, N$. O vizinho mais próximo é armazenado em vmp .

B. ENNS

Na busca pelo vetor com menor distância euclidiana quadrática (*squared euclidian distance*, SED), ou seja, o vetor-código mais próximo para um dado vetor de entrada, é utilizada sua correlação com a distância média quadrática (*squared*

mean distance, SMD), $d_{M,i}$, definida como [17]

$$d_{M,i} = \left[\left(\sum_{j=1}^K X_j - \sum_{j=1}^K W_{ij} \right)^2 \right] / K. \quad (10)$$

Uma importante observação em imagens reais é que o vetor-código de menor SED normalmente está na vizinhança do vetor de menor SMD, que é o vetor-código mais próximo de um vetor de entrada nos termos da medida SMD. Isto, porque a diferença média entre a soma das componentes W_{ij} de um vetor-código \vec{W}_i e das componentes X_j de um vetor de entrada \vec{X} é, por si só, uma estatística suficientemente próxima da distância Euclidiana [17]. Então, se a busca pelo vetor de menor SED, o qual representa o vetor mais próximo, iniciar pelo vetor com menor SMD e proceder para o seguinte mais próximo no sentido da medida SMD, existirá uma alta probabilidade de encontrar o vetor de menor SED em poucas iterações.

Uma vez que o vetor mais próximo é encontrado, a seguinte inequação, verdadeira para todo vetor-código \vec{W}_i , facilita o término da busca [17]:

$$d_{M,i} \leq d(\vec{X}, \vec{W}_i). \quad (11)$$

Qualquer dos próximos vetores-código em que a distância média quadrática for maior que a menor distância euclidiana quadrática já registrada pode ser seguramente rejeitado sem necessidade de calcular a distância euclidiana destes vetores, pois devido à Inequação (11), a distância deles certamente é maior que a menor distância:

$$d_{min} < d_{M,i} \leq d(\vec{X}, \vec{W}_i), \quad (12)$$

em que d_{min} é a menor distância já encontrada. Desta forma, caso a condição (13) seja verdadeira, o vetor-código \vec{W}_i pode ser eliminado da busca, em que m_i é a média do vetor-código \vec{W}_i e m_x a média do vetor de treino \vec{X} [17].

$$m_i \geq m_x + \sqrt{d_{min}/K} \text{ ou } m_i \leq m_x - \sqrt{d_{min}/K} \quad (13)$$

Caso contrário, é aplicado o algoritmo PDS para calcular a distorção e atualizar o valor de d_{min} . A busca é realizada para cima e para baixo no conjunto de vetores-código, que devem estar ordenados de forma crescente por suas médias, partindo do menor SMD encontrado previamente. Caso a condição dada em (13) seja satisfeita para alguma das direções, a busca continua apenas na outra direção até o vetor-código mais próximo ser encontrado [20].

A estrutura básica do algoritmo ENNS [17] é apresentada na Fig. 2 para a codificação de um dado vetor de treino \vec{X} . Pressupõe-se que os vetores do dicionário já estão em ordem crescente pelas médias. O vizinho mais próximo é armazenado em vmp .

A Fig. 3 ilustra um exemplo do procedimento de busca. Assumindo que \vec{W}_i é o vetor-código com SMD mínimo, o algoritmo procede a busca pela seus vizinhos de índice menor e maior até a condição da Expressão (13) ser satisfeita pela primeira vez no $(i - 2)$ -ésimo vetor. O restante dos vetores de índice mais alto são todos rejeitados e a busca continua apenas nos vetores-código de menor índice. O teste tem sucesso

```

1: Encontrar  $i$  tal que  $d_{M,i}$  é mínimo
2:  $d_{min} \leftarrow d(\vec{X}, \vec{W}_i)$ 
3:  $vmp \leftarrow i$ 
4:  $up \leftarrow down \leftarrow True$ 
5:  $d \leftarrow u \leftarrow i$ 
6: while  $up = True$  or  $down = True$  do
7:   if  $down = True$  then
8:      $d \leftarrow d + 1$ 
9:     if  $d > N$  or  $Mean(\vec{W}_d) \geq Mean(\vec{X}) + \sqrt{d_{min}/K}$ 
10:    then
11:       $down \leftarrow False$ 
12:    else
13:       $d_{min} \leftarrow d(\vec{X}, \vec{W}_d)$ 
14:       $vmp \leftarrow d$ 
15:    end if
16:  end if
17:  if  $up = True$  then
18:     $u \leftarrow u - 1$ 
19:    if  $u < 1$  or  $Mean(\vec{W}_u) \leq Mean(\vec{X}) - \sqrt{d_{min}/K}$ 
20:    then
21:       $up \leftarrow False$ 
22:    else
23:       $d_{min} \leftarrow d(\vec{X}, \vec{w}_u)$ 
24:       $vmp \leftarrow u$ 
25:    end if
26:  end if
27: end while

```

Figura 2: Passo a passo do algoritmo ENNS.

novamente no $(i - 3)$ -ésimo vetor, indicando que não há nenhum outro vetor mais próximo que o mais próximo já registrado.

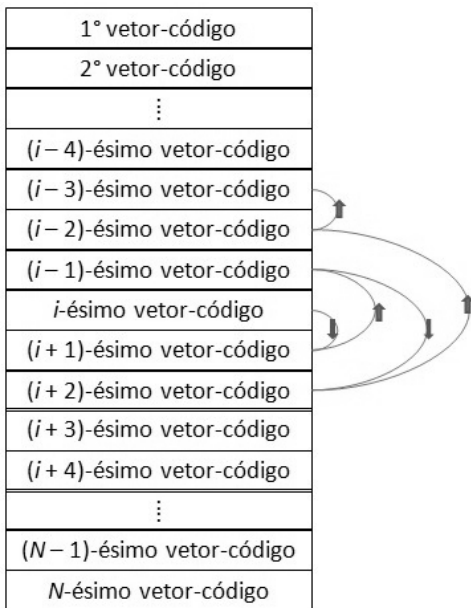


Figura 3: Um exemplo da busca realizada pelo algoritmo ENNS.

C. EENNS

Dois vetores com o mesmo valor de média ainda podem ter uma longa distância entre eles. Baseado nisto, o algoritmo EENNS [18] introduz outra condição significativa para rejeitar mais vetores-códigos, o desvio padrão. Assuma v_x e v_i como o desvio padrão do vetor de treino \vec{X} e do vetor-código \vec{W}_i , respectivamente, então

$$(v_x - v_i)^2 \leq d(\vec{X}, \vec{W}_i). \quad (14)$$

Com esta inequação, pode-se obter o seguinte critério de eliminação:

$$v_i \geq v_x + \sqrt{d_{min}} \text{ ou } v_i \leq v_x - \sqrt{d_{min}}. \quad (15)$$

O algoritmo EENNS consiste em dois passos. No primeiro é verificada se a condição da Expressão (13) do algoritmo ENNS é satisfeita. Caso o vetor-código não seja rejeitado, no segundo passo, é aplicada a condição dada em (15) e somente então, caso não seja verdadeira, é calculada a distância utilizando o algoritmo PDS.

D. EEENNS

Finalmente, o algoritmo EEENNS [19] utiliza as médias, desvios padrão e normas dos vetores como três mecanismos para acelerar o processo de busca do vizinho mais próximo.

Assumindo $\|\vec{X}\|$ e $\|\vec{W}_i\|$ as normas do vetor de treino \vec{X} e do vetor-código \vec{W}_i respectivamente, tem-se que

$$(\|\vec{X}\| - \|\vec{W}_i\|)^2 \leq d(\vec{X}, \vec{W}_i). \quad (16)$$

Com esta inequação, pode-se obter o seguinte critério de eliminação:

$$\|\vec{W}_i\| \geq \|\vec{X}\| + \sqrt{d_{min}} \text{ ou } \|\vec{W}_i\| \leq \|\vec{X}\| - \sqrt{d_{min}} \quad (17)$$

O algoritmo EEENNS, então, consiste em três passos. No primeiro e no segundo são aplicadas as comparações conforme o algoritmo EENNS. Caso o vetor-código não seja rejeitado, no terceiro passo, é aplicada a condição dada em (17) e então, caso também não seja verdadeira, é calculada a distância utilizando o algoritmo PDS.

V. RESULTADOS

Esta seção apresenta os resultados de simulação obtidos pelos algoritmos implementados, utilizando a imagem Lena (apresentada na Fig. 4) de 256×256 pixels com 256 níveis de cinza. A imagem foi utilizada como conjunto de treino na etapa de projeto de dicionários e como imagem de transmissão. Na etapa de projeto de dicionários para o PSO-COVQ foram utilizados 50 dicionários iniciais aleatórios (composto por vetores-código escolhidos de forma aleatória do conjunto de treino). Foram avaliados projetos de dicionários de tamanho $N = 32, 64, 128$ e 256 , com dimensão $K = 16$ (blocos de 4×4 pixels), e com limiar de distorção $\mu = 0,001$, projetados para transmissão de imagens por canal binário simétrico com probabilidade de erro de bit (V) de 0,005; 0,010; 0,050 e 0,100.

Foi utilizado um computador com Windows 7, processador Intel Core i7-4500U e 8 GB de memória RAM. A



Figura 4: Imagem *Lena* de 256×256 pixels.

implementação foi realizada na linguagem de programação C++.

O parâmetro cognitivo da partícula (c_1) e de aprendizado social (c_2) foram obtidos a partir de uma análise prévia, em que foram observados valores para c_1 e c_2 que produzissem um ambiente mais competitivo entre as partículas, procurando obter um número maior de troca de partículas escolhidas ao longo do processo [11]. Os valores utilizados nas simulações são respectivamente 0, 5 e 0, 7.

A Tabela I apresenta o tempo médio do projeto de dicionários utilizando os algoritmos PDS, EEENNS e Busca Total (BT) ou Busca Exaustiva após 10 simulações (cada uma com 50 dicionários iniciais diferentes).

Conforme se pode observar, a técnica mais eficiente é o algoritmo EEENNS, que apresenta uma redução de tempo médio de simulação de até 94, 3% para $N = 256$ e $V = 0, 005$, e 92, 6% para $N = 128$ e $V = 0, 010$ quando comparado à busca total.

Tabela I: Tempo médio (s) de 10 simulações do projeto de dicionários PSO-COVQ

V	Método	N			
		32	64	128	256
0,005	BT	1029,36	3858,04	11621,36	63390,07
	PDS	920,97	2374,88	7532,74	22168,04
	EEENNS	139,54	412,61	1088,01	3610,52
0,010	BT	868,71	4337,65	14119,98	44774,75
	PDS	545,78	1994,02	5668,99	22431,16
	EEENNS	132,72	476,35	1046,68	3034,31
0,050	BT	687,56	2395,99	11093,33	60924,06
	PDS	436,74	1329,88	5699,94	20753,40
	EEENNS	120,96	389,07	1523,25	5335,86
0,100	BT	512,41	1449,11	10221,66	41394,30
	PDS	287,68	790,09	3065,59	15952,10
	EEENNS	103,29	236,11	998,91	4318,40

A avaliação da qualidade de imagens pode ser realizada com uso da relação sinal-ruído de pico (PSNR, *Peak Signal to Noise Ratio*). Neste trabalho, avaliou-se o valor médio de PSNR das imagens reconstruídas, considerando 10 simulações (cada uma com 50 dicionários iniciais diferentes) e 20 transmissões através de um canal binário simétrico.

Os dois algoritmos de aceleração utilizados não introduzem erro ao projeto de dicionários, portanto a qualidade final da imagem reconstruída não deve ser afetada. Porém o algoritmo

Tabela II: Valores Médios de PSNR de imagens *Lena* reconstruídas com projeto de dicionários PSO-COVQ

V	Método	N			
		32	64	128	256
0,005	BT	25,36	26,13	26,31	26,70
	PDS	25,39	26,14	26,34	26,72
	EEENNS	25,31	26,05	26,30	26,67
0,010	BT	24,74	24,73	24,63	24,89
	PDS	24,79	24,73	24,69	24,98
	EEENNS	24,73	24,79	24,67	24,97
0,050	BT	20,75	20,97	21,13	21,04
	PDS	20,71	20,93	21,14	20,90
	EEENNS	20,72	20,97	21,20	20,94
0,100	BT	19,09	19,09	18,88	19,00
	PDS	19,11	19,08	18,97	18,97
	EEENNS	19,08	19,08	18,97	18,95

PSO utiliza um fator de aleatoriedade (r_1 e r_2) que diferencia os dicionários finais gerados. Esta diferença pode ser visualizada na Tabela II, que apresenta o valor do PSNR médio obtido com uso dos algoritmos da busca total, PDS e EEENNS. Nota-se que a maior diferença encontrada é de 0, 14 dB para $N = 128$ e $V = 0, 050$ com relação aos algoritmos PDS e BT, o que representa um desvio de apenas 0, 7%. A Fig. 5 apresenta as imagens reconstruídas, neste caso, para os três algoritmos utilizados.



(a) Busca Total



(b) PDS



(c) EEENNS

Figura 5: Imagens *Lena* reconstruídas para $N = 128$ e $V = 0, 050$.

VI. CONCLUSÃO

Este artigo apresentou a aplicação de técnicas de aceleração da busca pelo vizinho mais próximo, tradicionalmente usadas na etapa de codificação por distância mínima da QV, ao projeto de dicionários com otimização por enxame de partículas para quantização vetorial otimizada para canal. Os métodos,

chamados PDS e EEENNS, quando aplicados ao projeto de dicionários PSO-COVQ, possibilitaram acelerar o projeto de dicionários para quantização vetorial otimizada para canal. A economia de tempo de projeto foi obtida por meio da redução do número de operações lógicas e aritméticas da etapa de busca pelo vizinho mais próximo.

Os resultados obtidos mostram que é possível alcançar uma melhora sobre o algoritmo convencional de busca exaustiva em relação à complexidade computacional da etapa de particionamento do projeto de dicionários do PSO-COVQ, sem afetar a qualidade final da imagem, utilizando os algoritmos PDS e EEENNS. O último apresenta superioridade com relação à busca exaustiva pelo vizinho mais próximo e em relação ao PDS, reduzindo o tempo médio em até 94,3% quando comparado ao projeto com a busca exaustiva.

Como trabalhos futuros podem ser citados a investigação da influência da quantidade de partículas no valor do PSNR das imagens reconstruídas e a avaliação de outras técnicas de aceleração da busca do vizinho mais próximo.

AGRADECIMENTOS

Os autores expressam os agradecimentos ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e à Fundação de Amparo à Ciência e Tecnologia do Estado de Pernambuco (FACEPE) pelo apoio financeiro.

REFERÊNCIAS

- [1] A. Gersho e R. M. Gray, "Vector quantization and signal compression", *Kluwer Academic Publishers*, Boston, MA, 1992.
- [2] R. M. Gray, "Vector quantization", *IEEE ASSP Magazine*, vol. 1, pp. 4-29, 1984.
- [3] D. Tsolakis, G. Tsekouras e J. Tsimikas, "Fuzzy vector quantization for image compression based on competitive agglomeration and a novel codeword migration strategy", *Engineering Applications of Artificial Intelligence*, vol. 25, pp. 1212-1225, 2012.
- [4] D. Tsolakis, G. Tsekouras, A. Niros e A. Rigos, "On the systematic development of fast fuzzy vector quantization for grayscale image compression" *Neural Networks*, vol. 36, pp. 83-96, 2012.
- [5] Y. Qian, "Antenna location design for distributed antenna systems with selective transmission", *International Conference on Wireless Communication & Signal Processing*, pp. 1-5, 2009.
- [6] S. M. Hosseine e A. Naghsh-Nilchi, "Medical ultrasound image compression using contextual vector quantization", *Computers in Biology and Medicine*, vol. 42, pp. 743-750, 2012.
- [7] S.-R. Tsui e C.-T. Huang, "A new adaptive steganographic method based on gradient adjacent prediction and side-match vector quantization", *Journal of Information Hiding and Multimedia Signal Processing*, vol. 4, pp. 215-224, 2013.
- [8] S. Nakaya e S. Wada, "Audio watermarking based on vector quantization index modulation using audio fingerprint", *Electronics and Communications in Japan*, vol. 98, pp. 14-23, 2015.
- [9] F. Madeiro e W. T. A. Lopes, "Introdução à compressão de sinais", *Revista de Tecnologia da Informação e Comunicação*, no. 1, pp. 33-40, 2011.
- [10] N. Farvardin, "A study of vector quantization for noisy channels", *IEEE Transactions on Information Theory*, vol. 36, pp. 799-809, 1990.
- [11] H. A. S. Leitão, W. T. A. Lopes e F. Madeiro, "PSO algorithm applied to codebook design for channel-optimized vector quantization", *IEEE Latin America Transactions*, vol. 13, pp. 961-967, 2015.
- [12] W. T. A. Lopes, F. Madeiro, J. F. Galdino, B. G. Aguiar Neto e M. S. Alencar, "Diversidade em modulação aplicada a canais de comunicações móveis: efeito dos erros de estimação de canal na transmissão de imagens", *Revista da Sociedade Brasileira de Telecomunicações*, vol. 19, pp. 22-33, 2004.
- [13] W. T. A. Lopes, F. Madeiro, B. G. Aguiar Neto e M. S. Alencar, "Combining modulation diversity and index assignment to improve image VQ for Rayleigh fading channel", *Learning and Nonlinear Models*, Brasil, vol. 2, pp. 22-33, 2004.
- [14] N. Favardin e V. Vaishampayan, "On the performance and complexity of channel-optimized vector quantizers", *IEEE Transactions on Information Theory*, vol. 37, pp. 155-160, 1991.
- [15] J. Kennedy e R. C. Eberhart, "Particle swarm optimization", *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942-1948, 1995.
- [16] C. D. Bei e R. M. Gray, "An improvement of the minimum distortion encoding algorithm for vector quantization", *IEEE Transactions on Communications*, vol. 33, pp. 1132-1133, 1985.
- [17] S.-W. Ra e J.-K. Kim, "Fast mean-distance-ordered partial codebook search algorithm for image vector quantization", *IEEE Transactions on Circuits Systems II*, vol. 40, pp. 576-579, 1993.
- [18] C.-H. Lee e L.-H. Chen, "Fast closest codeword search algorithm for vector quantization", *IEE Proceedings - Vision, Image and Signal Processing*, vol. 141, pp. 143-148, 1994.
- [19] Z. M. Lu e S. H. Sun, "Equal-average equal-variance equal-norm nearest neighbor search algorithm for vector quantization", *IEICE Transactions on Information and Systems*, vol. E-86D, pp. 660-663, 2003.
- [20] S. C. Chu, Z. M. Lu e J. S. Pan, "Hadamard transform based fast codeword search algorithm for high-dimensional VQ encoding", *Information Sciences*, vol. 177, pp. 734-746, 2007.