

Casamento Lateral Generalizado Combinado à Recorrência de Padrões Multiescalas: Um Novo Esquema para a Compressão de Imagens

Eddie B. L. Filho, Úrsula Abecassis, Waldir S. S. Júnior, Eduardo A. B. da Silva e Murilo B. de Carvalho

Resumo—O SM-MMP (*Side-Match Multidimensional Multiscale Parser*) é um método baseado no casamento aproximado de padrões multiescalas, no qual o dicionário é construído considerando-se restrições de suavidade nos limites dos blocos em codificação, o que resulta em um melhor desempenho para imagens suaves. Neste artigo, é apresentada uma generalização onde o casamento com blocos vizinhos é realizado de forma hierárquica e com um maior grau de restrição, e do qual o SM-MMP pode ser considerado um caso particular. Além disso, uma nova estratégia de utilização de dicionário é apresentada, a qual emprega os elementos da vizinhança causal do bloco em codificação. As simulações realizadas em imagens em níveis de cinza mostram que o esquema proposto é eficaz, apresentando um desempenho superior ao do seu antecessor.

Palavras-Chave—Casamento de Padrões Recorrentes, Decomposição Multiescalas, Compressão de Sinais Multidimensionais, Quantização Vetorial, Side-Match, Three-Sided Side-Match.

Abstract—The SM-MMP (*Side-Match Multidimensional Multiscale Parser*) is a method based on approximate multiscale pattern matching, where the dictionary is built considering smoothness constraints around block boundaries, which results in a better performance for smooth images. In this work, a generalized framework is presented, in which the match attempt with neighboring blocks is performed in a hierarchical way and with a greater degree of restriction. The SM-MMP can be considered as a particular case of this new approach. Besides, a new dictionary usage strategy is presented, which employs elements from the causal neighborhood of the block being encoded. The simulations performed on grayscale images show that the proposed framework is effective, presenting a superior performance when comparing to its predecessor.

Keywords—Recurrent Pattern Matching, Multiscale Decomposition, Multidimensional Signal Compression, Vector Quantization, Side-Match, Three-Sided Side-Match.

I. INTRODUÇÃO

O SM-MMP (*Side-Match Multidimensional Multiscale Parser*) [1], [2] é um algoritmo desenvolvido recentemente e baseado no MMP (*Multidimensional Multiscale Parser*) [3], sendo que este último consiste em um esquema de compressão com perdas baseado no casamento aproximado de padrões multiescalas. O SM-MMP processa os blocos de uma imagem de forma dependente, baseando-se em uma métrica de similaridade com relação às escolhas feitas para os blocos previamente codificados, selecionando elementos de um dicionário \mathcal{D} que possuam pixels de borda semelhantes aos dos seus vizinhos. O primeiro bloco (canto superior esquerdo) é codificado com o algoritmo MMP e os outros, varrendo-se a imagem de cima

para baixo e da esquerda para a direita, criando-se dicionários de estado \mathcal{D}_s com os N elementos que atendem ao critério de similaridade (também chamado de critério de continuidade), conhecido como rugosidade [1], [2]. O dicionário \mathcal{D}_s é então utilizado nos processos de otimização e codificação, dando origem a uma imagem reconstruída com um menor efeito de blocagem e maior *PSNR*. Assim como no MMP padrão, o sinal é codificado através de sua segmentação em blocos de tamanhos variáveis e posterior aproximação por expansões e contrações de vetores presentes em um dicionário \mathcal{D} , que é construído durante o processo de codificação, acrescentando-se concatenações de expansões e contrações de vetores previamente codificados.

Neste artigo, propõe-se uma abordagem mais geral para o problema da codificação com casamento lateral, ou seja, considerando-se restrições de semelhança com os vizinhos do bloco a ser codificado. Os blocos de imagem agora dividem-se em dois grupos: blocos do estado inicial \mathcal{X}^j e blocos com casamento lateral generalizado $\mathcal{X}^{j(GSM)}$. Os primeiros são codificados com o algoritmo MMP ou SM-MMP, numa organização pré-estabelecida e periodicamente alocados no *grid* de imagem; os últimos são codificados com base nos blocos de estado inicial, fazendo casamento lateral com os mesmos, com cada bloco em codificação tendo até três vizinhos. Sendo assim, cria-se uma hierarquia na codificação dos blocos de imagem, onde a acurácia do casamento aumenta ao se passar de um grupo de blocos para outro. O SM-MMP, por exemplo, pode ser encarado como um derivado do mesmo esquema, mas sem blocos de estado inicial e com até dois vizinhos para cada bloco com casamento lateral. Como resultado, mantém-se a predição realizada pelo SM-MMP e incorpora-se blocos de referência \mathcal{X}^j , que são codificados com um dicionário maior e evitam uma possível propagação de erros devido à restrição na escolha de elementos. Em conjunto com este esquema, utilizam-se também deslocamentos na vizinhança do bloco atual para a sua codificação, o que aumenta a diversidade de escolhas para o casamento e captura estruturas periódicas presentes na imagem. As simulações realizadas mostram que o esquema proposto é eficaz e melhora o desempenho anteriormente apresentado pelo SM-MMP, numa abordagem unificada e geral para o casamento de padrões multiescalas.

O restante deste artigo está organizado conforme descrito a seguir. Na seção II, a versão do MMP com critério de continuidade, conhecida como SM-MMP, é apresentada. Na seção III, propõe-se o esquema com casamento lateral generalizado, chamado de GSM-MMP (*Generalized Side-Match Multidimensional Multiscale Parser*), assim como o novo dicionário de vizinhança. A seção IV prevê resultados experimentais com imagens em níveis de cinza e a seção V apresenta as conclusões do trabalho.

II. O ALGORITMO SM-MMP

O SM-MMP é um algoritmo de compressão com perdas baseado no casamento aproximado de padrões multiescalas, que é uma extensão do já conhecido casamento aproximado

Eddie B. L. Filho^{*†}, Úrsula Abecassis[‡], Waldir S. S. Júnior^{†Δ}, Eduardo A. B. da Silva[†] e Murilo B. de Carvalho[◇]. ^{*}Centro de Ciência, Tecnologia e Inovação do Pólo Industrial de Manaus, Rua Salvador, 391, Adrianópolis, Manaus-AM, 69057-040, BRAZIL. ^ΔUniversidade Federal do Amazonas (UFAM) e Centro de Pesquisa e Desenvolvimento em Tecnologia Eletrônica e da Informação (CETELI) Av. Gen. Rodrigo Octávio Jordão Ramos, 3000, Manaus - AM, Brasil. [†]Universidade Federal do Rio de Janeiro, Cx. P. 68504, Rio de Janeiro-RJ, 21941-972, BRAZIL. [‡]Fundação Centro de Análise, Pesquisa e Inovação Tecnológica, Av. Gov. Danilo de Matos Areosa, 381, Dist. Indust., Manaus - AM, 69075-351, BRAZIL. [◇]TET/CTC, Universidade Federal Fluminense, Rua Passo da pátria, 156, Niterói - RJ, 24210-240, BRAZIL. E-mails: eddie@ctpim.org.br, ursula@fucapi.br, waldirjr@ufam.edu.br, eduardo@lps.ufjf.br, murilo@telecom.uff.br.

de padrões [3], no qual vetores de dimensões diferentes podem ser aproximados. No casamento aproximado de padrões multiescalas, dois vetores u e v de comprimentos diferentes, ou seja, $l(u) \neq l(v)$, podem ser aproximados. Isto é possível através de uma transformação de escala $T_N^M(x) : \mathcal{R}^N \mapsto \mathcal{R}^M$, que ajusta as dimensões dos vetores antes do casamento ser executado, como ilustrado na Figura 1.

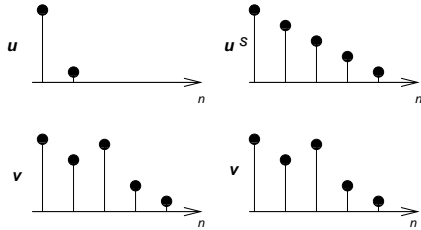


Fig. 1. Casamento aproximado de padrões multiescalas.

No SM-MMP bidimensional, há um dicionário $\mathcal{D} = \{b_0, b_1, \dots, b_{L-1}\}$ de blocos de tamanho fixo, que o mesmo utiliza para comprimir sub-blocos, de tamanho variável, de um bloco de entrada

$$X_{k,l}^0 = \begin{bmatrix} x(0,0) & \dots & x(0, N-1) \\ \vdots & \ddots & \vdots \\ x(M-1,0) & \dots & x(M-1, N-1) \end{bmatrix},$$

onde as dimensões $M \times N$ são potências de dois e k e l definem a localização do bloco na *grid* de imagem. Na tentativa de codificar $X_{k,l}^0$, o MMP procura, no dicionário de estado \mathcal{D}_s , o melhor bloco b_{i_0} para substituir $X_{k,l}^0$, sem se importar com as dimensões de cada b_{i_0} . A escolha é feita com base em algum critério de semelhança, como o erro quadrático $\xi_0 = \|X^0 - b_{i_0}^s\|^2$, onde $b_{i_0}^s$ é a versão escalonada de b_{i_0} . Se ξ_0 é menor que um dado limiar ξ_L , o bloco b_{i_0} é escolhido como a aproximação para $X_{k,l}^0$ e o algoritmo gera um *flag* '1', indicando um casamento bem sucedido, e o índice i_0 ; caso contrário, o bloco de entrada é particionado em dois outros blocos iguais. Se $M > N$, as partições são

$$X_{k,l}^1 = \begin{bmatrix} x(0,0) & \dots & x(0, N-1) \\ \vdots & \ddots & \vdots \\ x(M/2-1,0) & \dots & x(M/2-1, N-1) \end{bmatrix}$$

e

$$X_{k,l}^2 = \begin{bmatrix} x(M/2,0) & \dots & x(M/2, N-1) \\ \vdots & \ddots & \vdots \\ x(M-1,0) & \dots & x(M-1, N-1) \end{bmatrix}.$$

Entretanto, se $M \leq N$, as partições ficam

$$X_{k,l}^1 = \begin{bmatrix} x(0,0) & \dots & x(0, N/2-1) \\ \vdots & \ddots & \vdots \\ x(M-1,0) & \dots & x(M-1, N/2-1) \end{bmatrix}$$

e

$$X_{k,l}^2 = \begin{bmatrix} x(0, N/2) & \dots & x(0, N-1) \\ \vdots & \ddots & \vdots \\ x(M-1, N/2) & \dots & x(M-1, N-1) \end{bmatrix}.$$

Quando isto ocorre, o algoritmo gera um *flag* '0', indicando que houve uma divisão do bloco inicial, e repete o procedimento para $X_{k,l}^1$. Se um bloco semelhante a $X_{k,l}^1$ é encontrado no

dicionário \mathcal{D}_s , o algoritmo gera um *flag* '1' e um índice i_1 , passando então a processar $X_{k,l}^2$; caso contrário, um *flag* '0' é gerado e $X_{k,l}^1$ particionado. Este procedimento é recursivamente aplicado aos blocos, no sentido de construção de uma árvore binária, conforme ilustra a Figura 2 para um bloco X^0 de dimensões 4×4 .

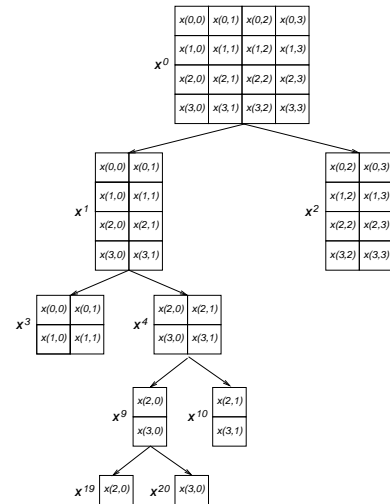


Fig. 2. Segmentação de um bloco 4x4 realizada pelo SM-MMP.

Ainda observando a Figura 2, é possível perceber que o particionamento dos blocos define uma árvore de segmentação \mathcal{S} , onde cada nó n_j está associado a um sub-bloco $X_{k,l}^j$. A árvore de segmentação associada à Figura 2 está ilustrada na Figura 3. Neste caso, a saída gerada pelo MMP seria 0, 0, 1, i_3 , 0, 0, i_{19} , i_{20} , 1, i_{10} , 1, i_2 . Como i_{19} e i_{20} pertencem à última escala possível (1×1), o *flag* '1' não é necessário. Os índices e *flags* resultantes são então codificados por um codificador aritmético [4], de modo a explorar a redundância existente entre os elementos do dicionário e *flags* através das diversas escalas. A seqüência de *flags* 0, 0, 1, 0, 0, 1, 1 define a árvore de segmentação e informa as dimensões do bloco $X_{k,l}^j$.

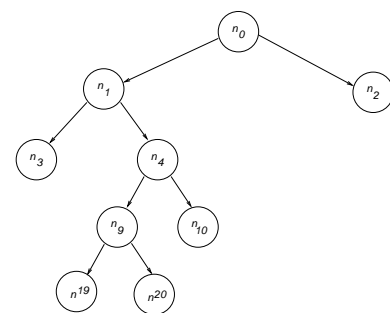


Fig. 3. Árvore de segmentação relativa à Figura 2.

Para um dado bloco $X_{k,l}^j$ de dimensões $M \times N$ associado ao nó n_j da árvore de segmentação \mathcal{S} , define-se o *vizinho superior* $U_{k,l}^j$ como o bloco de dimensões $N \times M$ imediatamente acima de $X_{k,l}^j$, e o *vizinho esquerdo* $L_{k,l}^j$ como o bloco de dimensões $N \times M$ imediatamente à esquerda de $X_{k,l}^j$, como ilustrado na Figura 4. O dicionário de estado \mathcal{D}_s utilizado para se codificar cada bloco $X_{k,l}^j$ é formado a partir do dicionário principal \mathcal{D} , modificando-se a probabilidade de ocorrência de cada elemento do dicionário de acordo com um critério de continuidade [1], [2], [5], [6], [7], [8], [9]. Todos os elementos

do dicionário que não obedecem ao mesmo são excluídos da codificação (recebem probabilidade de ocorrência zero), colocando-se os restantes em \mathcal{D}_s . A decisão de inclusão ou não de um dado elemento é baseada na medida de rugosidade $r_{ij} = R(\mathbf{U}^j, \mathbf{L}^j, \mathbf{b}_i)$, definida em [2].

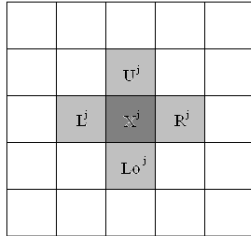


Fig. 4. Vizinhos do bloco em codificação.

\mathcal{D}_s é composto pelos N elementos \mathbf{b}_i do dicionário \mathcal{D} que possuem as menores medidas de rugosidade r_{ij} . O tamanho N do dicionário de estado depende do nível de *atividade* dos blocos \mathbf{U}^j e \mathbf{L}^j . A atividade de um bloco \mathbf{X} de dimensões $M \times N$ é avaliada através da função $A(\mathbf{X})$, definida em [2].

O tamanho N é determinado através do cálculo da razão $N_{max}((A(\mathbf{U}^j) + A(\mathbf{L}^j))/2)A_{max}^{-1}$, onde A_{max} é o maior valor de $A(\mathbf{X})$ dentre os obtidos para todos os blocos de entrada da imagem e N_{max} é o tamanho máximo permitido para o dicionário de estado.

O dicionário principal é atualizado através da adição de versões expandidas e contraídas de blocos previamente codificados, de maneira semelhante ao que é feito nos codificadores Lempel-Ziv [10]. Por exemplo, logo após a escolha dos blocos $b_{i_{19}}$ e $b_{i_{20}}$ na Figura 2, os mesmos podem ser escalonados e concatenados para formar \tilde{X}^9 , o valor reconstruído para X^9 , ou seja,

$$\tilde{X}^9 = \begin{bmatrix} \tilde{x}(2,0) \\ \tilde{x}(3,0) \end{bmatrix},$$

incluindo-o no dicionário. Como X^9 já havia sido particionado, o que significa que sua aproximação não foi obtida no dicionário \mathcal{D} , a mesma não existe e pode ser adicionada para utilização futura.

Sabe-se que se $X_{k,l}^0$ possuir dimensões $M \times N$, com $M = N$, o processo de partição criará blocos com as seguintes dimensões: $\{M \times N, M \times N/2, M/2 \times N/2, M/2 \times N/4, M/2 \times N/4, \dots, 1 \times 1\}$, ou seja, $1 + \log_2(M) + \log_2(N)$ dimensões diferentes. Para se diminuir a complexidade computacional do método e evitar a realização da transformação de escala T_N^M antes de cada tentativa de casamento, é necessário manter $1 + \log_2(M) + \log_2(N)$ cópias do dicionário, realizando T_N^M apenas na inclusão de cada elemento.

A árvore de segmentação \mathcal{S} pode ser otimizada num sentido taxa-distorção, permitindo uma distribuição dos bits para a codificação que leva em consideração as necessidades globais de cada bloco de entrada. A otimização começa a partir do bloco $X_{k,l}^0$ e particiona recursivamente os blocos $X_{k,l}^{2j+1}$, calculando os custos Lagrangeanos $J(n_{2j+1})$ [11] associados aos mesmos. A ordem de varredura dos elementos está representada na Figura 5. As estatísticas dos elementos $b_{i_{2k+1}}$ e *flags* utilizados, para todas as escalas $(p, q) = (0, 0), (0, 1), (1, 1), (1, 2), \dots, (\log_2(M), \log_2(N))$, são continuamente atualizadas, levando-se em consideração a utilização ou exclusão de um dado elemento. Quando o algoritmo chega à última escala, ou seja, 1×1 , o mesmo retorna analisando os custos das sub-árvores com início nos nós-filhos $\mathcal{S}(n_{2j+1})$ e $\mathcal{S}(n_{2j+2})$ e o custo do nó-pai n_j para decidir sobre a remoção das sub-árvores, o que acontece

se $J(n_j) \leq J(\mathcal{S}(n_{2j+1})) + J(\mathcal{S}(n_{2j+2}))$. Se o algoritmo decide pela remoção das sub-árvores, todas as atualizações devidas a $\mathcal{S}(n_{2j+1})$ e $\mathcal{S}(n_{2j+2})$ são removidas do dicionário \mathcal{D} e as estatísticas dos elementos e *flags* são corrigidas. O procedimento continua através da partição e análise dos outros sub-blocos $X_{k,l}^{2j+2}$ da partição atual (o bloco à direita ou abaixo) e então para os blocos seguintes à esquerda e acima, até que não haja mais elementos.

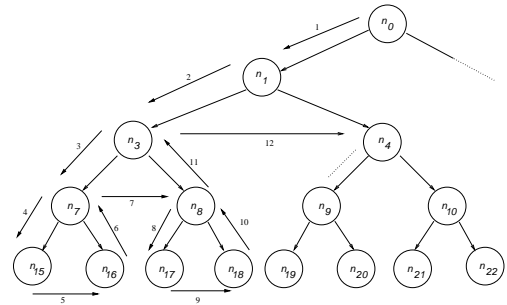


Fig. 5. Ordem de processamento dos nós durante a otimização no SM-MMP.

Para melhorar o desempenho do algoritmo, além das atualizações já mencionadas, inclui-se também versões deslocadas, na vizinhança causal, do bloco atual.

III. O ESQUEMA PARA O CASAMENTO LATERAL GENERALIZADO: GSM-MMP

A. O algoritmo base

No SM-MMP, cada bloco de sinal passa a ser codificado dependentemente dos seus vizinhos causais. Como os blocos de imagem $X_{k,l}^0$ são varridos no sentido de leitura, ou seja, da esquerda para a direita e de cima para baixo, há apenas dois vizinhos causais disponíveis: $\mathbf{U}_{k,l}^0 = X_{k-1,l}^0$ e $\mathbf{L}_{k,l}^0 = X_{k,l-1}^0$. Entretanto, se o modo como os blocos de imagem são varridos for modificado [12], mais vizinhos podem ser utilizados, como ilustrado na Figura 4. Define-se, então, o *vizinho inferior* $\mathbf{L}_{o,k,l}^0 = X_{k+1,l}^0$ como o bloco de dimensões $M \times N$ imediatamente abaixo de $\mathbf{X}_{k,l}^0$, e o *vizinho direito* $\mathbf{R}_{k,l}^0 = X_{k,l+1}^0$ como o bloco de dimensões $M \times N$ imediatamente à direita de $\mathbf{X}_{k,l}^0$.

Para que os novos vizinhos sejam utilizados, o presente esquema passará a classificar os blocos de imagem em dois grupos: blocos de estado inicial $\mathcal{X}_{k,l}^0$ e blocos com casamento lateral generalizado $\mathcal{X}_{k,l}^{0(GSM)}$. O primeiro grupo consiste nos primeiros blocos $\mathcal{X}_{k,l}^0 = X_{2k,2l}^0$ a serem codificados, que são organizados num esquema similar a uma subamostragem de 2:1 na horizontal e na vertical, conforme ilustra a Figura 6.

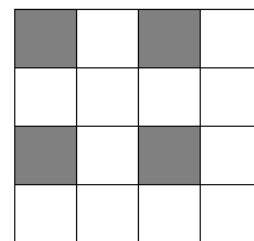


Fig. 6. Organização dos blocos de estado inicial.

Estes blocos podem ser codificados com o MMP original ou o SM-MMP. Entretanto, para melhorar o desempenho em taxas mais baixas e criar uma hierarquia de casamento lateral, todos os blocos de estado inicial são codificados utilizando-se

o SM-MMP (o tamanho de cada \mathcal{D}_s é quadruplicado, devido ao aumento da incerteza), com exceção de $\mathcal{X}_{0,0}^0$, que é o bloco fundamental e não possui qualquer vizinho, sendo necessária a utilização do MMP. Deste modo, o casamento lateral não é mais realizado com os vizinhos imediatamente acima e à esquerda, mas sim com $\mathcal{U}_{k,l}^0 = \mathcal{X}_{2k-2,2l}^0$ e $\mathcal{L}_{k,l}^0 = \mathcal{X}_{2k,2l-2}^0$, conforme ilustrado na Figura 7. Todas as características do algoritmo SM-MMP são utilizadas, inclusive os dicionários de estado \mathcal{D}_s . Esta estratégia melhora o desempenho do esquema em taxas baixas ($< 0,3$ bpp). Isto pode ser entendido levando em consideração o fato de que, se apenas o MMP fosse empregado na codificação destes blocos, dado que o mesmo não utiliza dicionários de estado, necessitaria de taxas \mathcal{R}_i para codificar os índices b_i e \mathcal{R}_t para codificar os *flags* descritores da árvore de segmentação muito maiores. Como pode ser facilmente observado, os blocos de estado inicial correspondem a 25% do total de blocos da imagem.

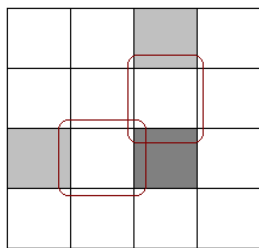


Fig. 7. Casamento lateral realizado nos blocos de estado inicial.

Após a codificação dos blocos $\mathcal{X}_{k,l}^0$, o *grid* inicial está gerado, com blocos-chave periodicamente espalhados pela imagem. Estes blocos podem ser utilizados como novos vizinhos para o casamento lateral, consistindo em uma nova formação do dicionário de estado para o processo de codificação. O próximo grupo de blocos, conhecidos como blocos com casamento lateral generalizado $\mathcal{X}_{k,l}^{0(GSM)} = \mathcal{X}_{2k/2k+1,2l+1/l}^0$, são codificados sob o conceito de casamento lateral generalizado, onde pode haver mais de dois vizinhos. Com a organização sugerida na Figura 6, podem ocorrer três situações (sem contar com as restrições dos blocos nas bordas da imagem, que são casos especiais), mostradas na Figura 8. Observa-se, então, que agora é possível haver casamento com até três vizinhos e de diferentes formas. A rugosidade e o tamanho do dicionário são calculados da mesma forma que no SM-MMP, mas com a adição de mais um elemento. Por exemplo, para o casamento tipo 3 mostrado na Figura 8, a rugosidade R e o tamanho do dicionário $L(D)$ seriam

$$\begin{aligned}
 R(\mathbf{U}^j, \mathbf{L}^j, \mathbf{b}_i) = & \sum_{k=0}^{N-1} \left| \left| \frac{U^j(M-2, k) - U^j(M-1, k) + b_i(0, k) - b_i(1, k)}{2} \right| \right. \\
 & \left. + b_i(0, k) - U^j(M-1, k) \right| \\
 & + \sum_{p=0}^{M-1} \left| \left| \frac{L^j(p, N-2) - L^j(p, N-1) + b_i(p, 0) - b_i(p, 1)}{2} \right| \right. \\
 & \left. + b_i(p, 0) - U^j(p, N-1) \right| \\
 & + \sum_{p=0}^{M-1} \left| \left| \frac{b_i(p, N-2) - b_i(p, N-1) + R^j(p, 0) - R^j(p, 1)}{2} \right| \right. \\
 & \left. + R^j(p, 0) - b_i(p, N-1) \right| \quad (1)
 \end{aligned}$$

e

$$L(D) = N_{max} \left(\left(A(\mathbf{U}^j) + A(\mathbf{L}^j) + A(\mathbf{R}^j) \right) / 3 \right) A_{max}^{-1}, \quad (2)$$

respectivamente. Extensões similares podem ser feitas para o casamento tipo 1. O dicionário \mathcal{D} , no GSM-MMP, é montado e atualizado da mesma maneira que em [2].

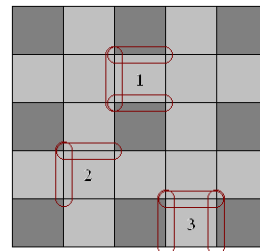


Fig. 8. Tipos de casamento encontrados em blocos com casamento lateral generalizado.

Com isso, os blocos utilizados na criação de \mathcal{D}_s são escolhidos com uma maior restrição e são mais casados com os seus vizinhos causais. Como já foi falado, tanto o SM-MMP quanto até mesmo o MMP original podem ser considerados casos especiais deste esquema.

A complexidade computacional do GSM-MMP é comparável a do SM-MMP, porém, o próprio algoritmo base MMP é mais complexo que alguns esquemas baseados em wavelets, como o JPEG2000 [13]. Uma análise da complexidade computacional do MMP pode ser encontrada em [14].

B. O dicionário de vizinhança

Apesar do GSM-MMP já utilizar um esquema de atualização que leva em consideração elementos da vizinhança causal do bloco [2], nem todos os deslocamentos possíveis são considerados. Além disso, o critério de continuidade pode não incluir em \mathcal{D}_s algum elemento interessante para a codificação de $X_{k,l}^j$, dada a incerteza com respeito aos pixels que não fazem parte da borda dos blocos [1]. Para propiciar um dicionário mais rico e capturar estruturas periódicas no sinal, resultando numa maior qualidade da imagem reconstruída, utiliza-se também o conceito de dicionário de vizinhança \mathcal{D}_N . Tal dicionário consiste em todos os deslocamentos possíveis, pixel-a-pixel, na vizinhança causal de $X_{k,l}^j$, como ilustrado na Figura 9. Para cada $X_{k,l}^j$, uma janela com as mesmas dimensões é deslocada na sua vizinhança causal, onde cada deslocamento corresponde a um índice de dicionário b_i^N , e procura-se o elemento que melhor aproxima $X_{k,l}^j$. Como resultado, quando certos blocos se repetem na imagem de forma determinística, alguns elementos b_i^N serão mais utilizados que outros, o que proporciona ganhos significativos quando do uso de um codificador aritmético [4].

IV. RESULTADOS DE SIMULAÇÕES

O SM-MMP e o GSM-MMP foram implementados em *linguagem C* e executados em ambiente *Linux*. Os programas gerados foram utilizados para compressão de imagens em níveis de cinza, inicialmente divididas em blocos $X_{k,l}^0$ de 16×16 . Os blocos de entrada foram processados em seqüência pelos algoritmos, no sentido de leitura, ou seja, da esquerda para a direita e de cima para baixo.

O dicionário inicial continha apenas vetores na escala de 1×1 e foi iniciado com $\mathcal{D}_0 = \{0, 4, 8, \dots, 252, 255\}$. Os

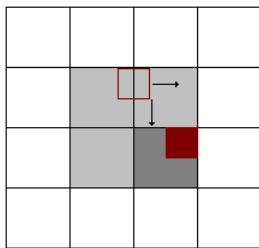


Fig. 9. Exemplificação do dicionário de vizinhança.

vetores, em todas as outras escalas, foram obtidos através de uma transformação de escala bidimensional separável, implementada utilizando-se as operações descritas em [2]. Com blocos iniciais de 16×16 , 9 escalas diferentes podem ser obtidas.

As Figuras 10, 11, 12, 13, 14 e 15 mostram o desempenho taxa-distorção dos algoritmos para as imagens Cameraman, Einstein, House, Lena, F-16 e PP1209, sendo as três primeiras de 256×256 e as outras de 512×512 pixels. Os resultados para o JPEG2000 [13] também são apresentados, possibilitando uma comparação imediata entre os esquemas de compressão. As imagens Lena e F-16 foram obtidas no site <http://sipi.usc.edu/database>, a Cameraman e a House em <http://iie.fing.edu.uy/ense/assign/codif/material.htm> e a Einstein em http://scien.stanford.edu/labsite/scien_test_imagens_videos.html. A imagem PP1209 foi digitalizada do IEEE *Transactions on Image Processing*, volume 9, número 7, de julho de 2000. A página escolhida foi a de número 1209, que dá nome a imagem, e contém um misto de imagens em níveis de cinza (duas versões comprimidas da Lena), texto, fórmulas e gráficos. As Figuras mostram que:

- 1) O SM-MMP apresentou desempenhos melhores que os do JPEG2000 para as imagens Cameraman e PP1209.
- 2) O SM-MMP apresentou desempenhos comparáveis aos do JPEG2000 para as imagens Einstein e F-16.
- 3) O SM-MMP apresentou desempenho superior ao do JPEG2000, em taxas mais baixas, para a imagem House.
- 4) O GSM-MMP apresentou desempenhos superiores aos do SM-MMP para todas as imagens.
- 5) O GSM-MMP apresentou melhores desempenhos que o JPEG2000 para quatro das imagens, com vantagens da ordem de $\approx 2,2$ dB a $0,5$ bpp para a PP1209 (misto de imagens em níveis de cinza, texto e gráficos), $\approx 1,8$ dB a $0,5$ bpp para a Cameraman, $\approx 0,35$ dB a $0,4$ bpp para a Einstein e $\approx 0,9$ dB a $0,4$ bpp para a house.
- 6) O GSM-MMP apresentou desempenho comparável ao do JPEG2000 para a imagem F-16, ganhando em taxas mais baixas.
- 7) O GSM-MMP não apresentou o mesmo desempenho que o JPEG2000 para a imagem Lena.

A Figura 16 mostra a imagem Cameraman original e as versões reconstruídas pelo GSM-MMP e pelo JPEG2000, a $0,3$ bpp. É fácil perceber que os detalhes da imagem são mais preservados pelo GSM-MMP, inclusive os edifícios no lado direito, que não podem ser completamente identificados na versão do JPEG2000. O rosto do *cameraman* também está mais nítido.

V. CONCLUSÕES

Neste artigo, foi proposto um novo esquema para a compressão de imagens utilizando-se casamento aproximado de padrões multiescalas e casamento lateral, que também pode ser aplicado a outros tipos de sinais. A abordagem apresentou um

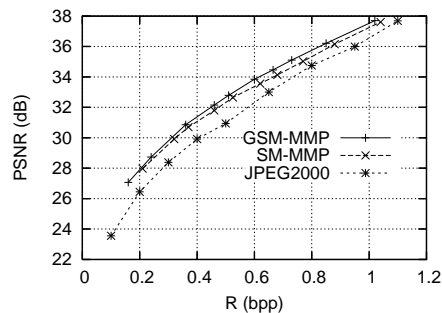


Fig. 10. Desempenho taxa-distorção para Cameraman 256×256 .

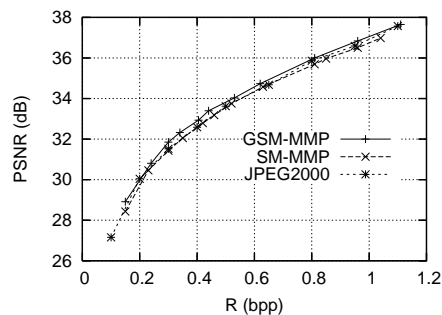


Fig. 11. Desempenho taxa-distorção para Einstein 256×256 .

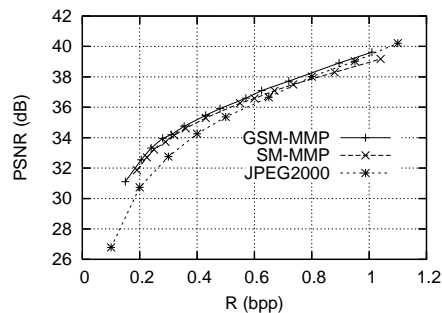


Fig. 12. Desempenho taxa-distorção para House 256×256 .

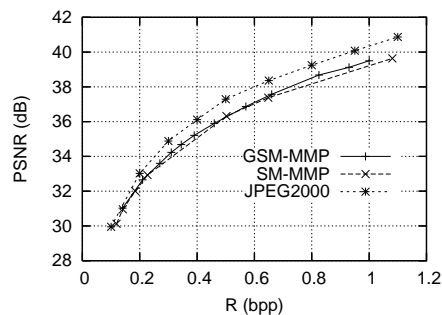


Fig. 13. Desempenho taxa-distorção para Lena 512×512 .

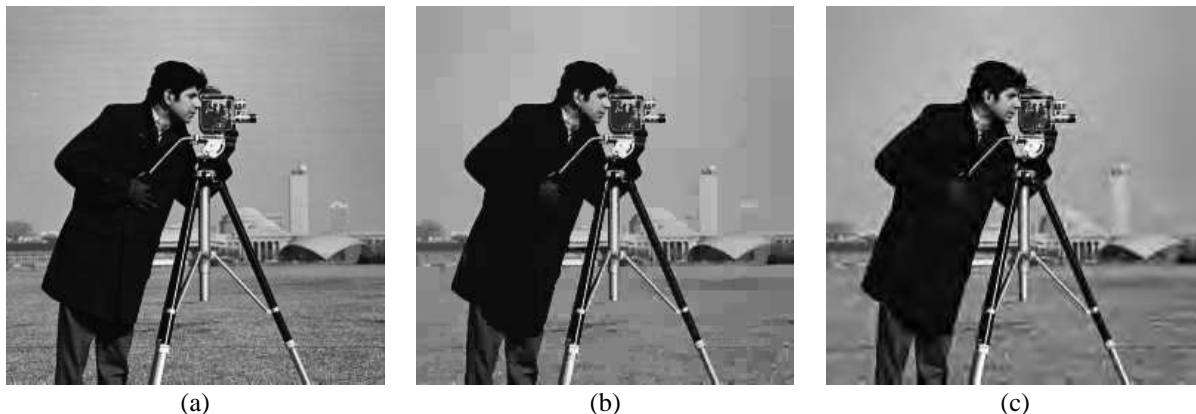


Fig. 16. Imagem Cameraman: (a) Imagem original. (b) GSM-MMP a 0,30 bpp. PSNR = 29,95 dB. (c) JPEG2000 a 0,30 bpp. PSNR = 28,30 dB.

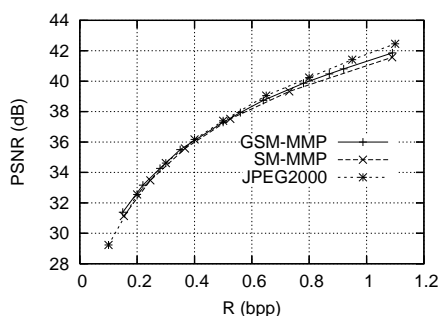


Fig. 14. Desempenho taxa-distorção para F-16 512 × 512.

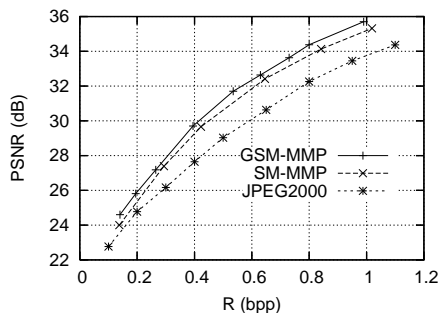


Fig. 15. Desempenho taxa-distorção para PP1209 512 × 512.

desempenho superior ao do seu predecessor SM-MMP, com ganhos significativos em taxas mais altas. O casamento lateral se tornou generalizado e hierárquico, dividindo os blocos de imagem em grupos que são comprimidos de maneiras diferentes e utilizando também uma nova estratégia de dicionário. Tanto o SM-MMP quanto o MMP podem ser considerados casos especiais do presente esquema, que pode ser simplificado ou modificado de acordo com a aplicação.

AGRADECIMENTOS

Este trabalho só foi possível devido ao apoio à pesquisa básica que o Centro de Tecnologia e Inovação do Pólo Industrial de Manaus – CT-PIM proporciona. Os autores gostariam de agradecer à Diretoria do CT-PIM, representada pelos Srs. Wesley Alves Pereira e Admilton Pinheiro Salazar, e também ao Centro Tecnológico do Exército, pelo suporte a este trabalho através financiamento FINEP/FAPEB, sob o convênio 2645/06.

REFERÊNCIAS

- [1] E. B. L. Filho, “Compressão de Imagens utilizando Recorrência de Padrões Multiescalas com Critério de Continuidade Inter-Blocos”, Masters Dissertation, COPPE/UFRJ, Rio de Janeiro, Brazil, April 2004.
- [2] E. B. L. Filho, M. B. de Carvalho, and E. A. B. da Silva, “Compressão de Sinais Multidimensionais utilizando Recorrência de Padrões Multiescalas com Critério de Continuidade Interblocos”, XXI Simpósio Brasileiro de Telecomunicações, Belém, Brazil, September 2004.
- [3] M. B. de Carvalho, E. A. B. da Silva and W. A. Finamore, “Multi-dimensional Signal Compression using Multiscale Recurrent Patterns,” Elsevier’s Signal Processing, Vol. 82, No. 11, pp. 1559-1580, November 2002.
- [4] I. H. Witten, R. M. Neal and J. G. Cleary, “Arithmetic Coding for Data Compression”, *Communications of the ACM*, Vol. 30, No. 6, pp. 520-540, June 1987.
- [5] T. Kim, “Side Match and Overlap Match Vector Quantizers for Images”, *IEEE Transactions on Image Processing*, Vol. 1, No. 2, pp. 170-185, February 1992.
- [6] Z. M. Lu, B. Yang and S. H. Sun, “Image Compression Algorithms Based on Side-Match Vector Quantizer with Gradient-Based Classifiers”, *IEICE Transactions on Information and Systems*, Vol. E85-D, No. 9, pp. 1409-1415, September 2002.
- [7] S. B. Yang and L. Y. Tseng, “Smooth Side-Match Classified Vector Quantizer with Variable Block Size”, *IEEE Transactions on Image Processing*, Vol. 10, No. 5, pp. 677-685, May 2001.
- [8] S. B. Yang, “General-Tree-Structured Vector Quantizer for Image Progressive Coding Using the Smooth Side-Match method”, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 13, No. 2, pp. 193-202, February 2003.
- [9] T. S. Chen and C. C. Chang, “A New Image Coding Algorithm Using Variable-Rate Side-Match Finite-State Vector Quantization”, *IEEE Transactions on Image Processing*, Vol. 6, No. 8, pp. 1185-1187, August 1997.
- [10] J. Ziv and A. Lempel, “Compression of individual sequences via variable-rate coding,” *IEEE Transactions on Information Theory*, Vol. it-24, No. 5, pp. 530-536, September 1978.
- [11] M. M. Denn, “Optimization by Variational Methods”, McGraw-Hill Book Company, 1969.
- [12] H. C. Wei, P. C. Tsai and J. S. Wang, “Three-Sided Side Match Finite-State Vector Quantization”, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 10, No. 1, pp. 51-58, February 2000.
- [13] D. S. Taubman, M. W. Marcellin, “JPEG2000: Image Compression Fundamentals, Standards, and Practice”, Kluwer Academic Publishers, 2001.
- [14] E. B. L. Filho, E. A. B. da Silva, M. B. Carvalho, W. S. S. Júnior and J. Koiller, “Electrocardiographic Signal Compression using Multiscale Recurrent Patterns”, *IEEE Transactions on Circuits and Systems-I: Regular Papers*, Vol. 52, No. 12, pp. 2739-2753, December 2005.