

# Compressão de Sinais Multidimensionais utilizando Recorrência de Padrões Multiescalas com Critério de Continuidade Interblocos

Eddie B. L. Filho, Murilo B. de Carvalho e Eduardo A. B. da Silva

**Resumo**—O MMP (Multidimensional Multiscale Parser) é um método baseado em recorrência de padrões multiescalas recentemente aplicado à compressão de imagens. Neste artigo, propõe-se uma extensão ao mesmo, chamada de SM-MMP (Side-match MMP), no qual o dicionário é construído considerando-se restrições de suavidade nos limites dos blocos em codificação, o que permite apresentar, quando a imagem processada é suave, resultados melhores que os do MMP. As simulações realizadas mostram que o método proposto é eficaz, apresentando melhoras de até 1,5 dB com relação ao MMP padrão para imagens em escala de cinza, ao passo que mantém o excelente desempenho para imagens com gráficos, texto ou mistas.

**Palavras-Chave**—Casamento de Padrões Recorrentes, Decomposição Multiescalas, Compressão de Sinais Multidimensionais, Quantização Vetorial, Side-match.

**Abstract**—The MMP (Multidimensional Multiscale Parser) is a method based on multiscale recurrent patterns recently applied for image compression. In this paper it is proposed an extension for it, the SM-MMP (Side-match MMP), in which the dictionary is built considering smoothness constraints around block boundaries. This allows it to perform better than MMP when the images are smooth. Indeed, the simulation results show that the proposed method is effective, yielding improvements of the order of 1.5 dB over the original MMP for grayscale images, while preserving the high performance of the original MMP for graphics, text and mixed images.

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

## I. INTRODUÇÃO

Em um trabalho recente [1], foi introduzido o algoritmo MMP (Multidimensional Multiscale Parser), que consiste num esquema de compressão com perdas baseado em casamento aproximado de padrões multiescalas. As simulações realizadas mostram que o MMP é bastante eficaz na compressão de um grande conjunto de imagens, superando, para imagens com texto, gráficos ou mistas, os codificadores que constituem o estado da arte na área de compressão de imagens. Apesar do desempenho do MMP para imagens suaves se mostrar melhor que o do JPEG [6], que é baseado em DCT, tal fato não se repete com relação aos codificadores baseados em DWT, como

o SPIHT [5] e o JPEG2000 [4]. Esta superioridade pode ser explicada através do fato de que o sucesso dos codificadores baseados em transformada reside na suposição de que a imagem em processamento é essencialmente de natureza passa-baixas. Isto significa que a maior parte da sua energia está concentrada em coeficientes de transformada de frequências mais baixas, resultando em esquemas de codificação eficientes para os coeficientes quantizados. Porém, essa suposição não é verdadeira para muitos tipos de imagens, como, por exemplo, aquelas contendo texto ou gráficos. Diferentemente da maioria dos métodos utilizados em compressão de imagens e vídeo, o MMP codifica uma dada fonte sem utilizar o paradigma de transformação-quantização-codificação de entropia, além de não fazer qualquer asserção com respeito às características do sinal a ser processado. Assim, embora comprima com boa qualidade um grande espectro de imagens, o MMP não apresenta, com relação aos codificadores baseados em DWT e para imagens suaves, resultados tão bons quanto os para imagens com texto, gráficos ou mistas. Baseando-se no que foi exposto, uma possibilidade para melhorar o desempenho do MMP em imagens mais suaves consiste em assumir algum modelo estatístico para a fonte e ajustar o algoritmo de acordo com os resultados. De preferência, isto deve ser feito de modo que o excelente desempenho em imagens de texto, gráficos ou mistas não seja afetada.

Neste artigo, propõe-se uma extensão ao MMP, chamada de SM-MMP (Side-match Multidimensional Multiscale Parser), cujo principal objetivo é melhorar o seu desempenho na codificação de imagens suaves. 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, que é construído durante o processo de codificação, acrescentando-se concatenações de expansões e contrações de vetores previamente codificados. Porém, diferentemente do MMP, o SM-MMP adota um modelo estatístico para a fonte. Ao codificar um dado bloco, o SM-MMP mede o comportamento da imagem na sua vizinhança causal, o que leva à construção de um subdicionário chamado de dicionário de estado, selecionando-se, do dicionário principal, vetores que obedecem a um dado critério de suavidade relativo à vizinhança causal. Tal dicionário pode ter um tamanho que vai de um mínimo pré-definido até o tamanho total do dicionário principal, o que é equivalente a atribuir, aos vetores que não obedecem ao critério de suavidade, probabilidade de ocorrência zero, que constitui uma asserção razoável para imagens suaves. Reduzindo o

Eddie B. L. Filho<sup>\*†</sup>, Murilo B. de Carvalho<sup>‡</sup> e Eduardo A. B. da Silva<sup>†</sup>,  
<sup>\*</sup>Genius Instituto de Tecnologia, Av. Açaí, 875, BL. E, Manaus - AM, 69075-904, BRASIL, <sup>†</sup>TET/CTC, Universidade Federal Fluminense, R. Passos da Pátria, 156, Niterói, RJ, 24210-240, BRASIL, <sup>‡</sup>PEE/COPPE/DEL/POLI, Universidade Federal do Rio de Janeiro, Cx. P. 68504, Rio de Janeiro - RJ, 21945-970, BRASIL, E-mails: efilho@genius.org.br, murilo@telecom.uff.br, eduardo@lps.uff.br. Este trabalho foi realizado no âmbito do convênio UFAM-UFRJ/COPPE, com financiamento da SUFRAMA.

dicionário principal para um dicionário de estado que contém apenas os elementos mais prováveis para a codificação de cada bloco, o SM-MMP tem o potencial de diminuir a taxa média para a codificação dos mesmos, ao passo que mantém a distorção média inalterada. Vale ressaltar que esta técnica é similar à utilizada no esquema de SMVQ (Side-Match Vector Quantization) [2], [3], [7], [8], [9], [10].

O restante deste artigo está organizado conforme descrito a seguir. Na seção II, a versão original do algoritmo MMP é apresentada. Na seção III, propõe-se a extensão com critério de suavidade, o SM-MMP. Resultados experimentais com imagens em níveis de cinza são apresentados na seção IV e a seção V apresenta as conclusões do trabalho.

## II. O ALGORITMO MMP

O algoritmo MMP é um esquema de compressão com perdas que utiliza a técnica de *casamento aproximado de padrões multiescalas*, uma extensão do já conhecido casamento aproximado de padrões [1], no qual vetores de dimensões diferentes podem ser aproximados. Isto é possível através de uma transformação de escala  $T_N^M : \mathbb{R}^N \mapsto \mathbb{R}^M$ , que ajusta as dimensões dos vetores antes do casamento ser executado [1]. No caso de imagens, utiliza-se uma transformação de escala bidimensional, que permite o casamento de dois blocos de imagem (matrizes) de dimensões diferentes. A transformação propriamente dita é decomposta em duas operações unidimensionais (transformação separável), transformando-se primeiro todas as linhas e depois todas as colunas, segundo as fórmulas (1) e (2). Estas fórmulas são aplicáveis a um vetor de  $N_0$  posições sendo transformado para um de  $N$  posições, ou seja, uma linha ou coluna do bloco por vez. Mais especificamente, a transformação de escala é implementada utilizando-se operações de mudança de taxa de amostragem.

Para que as dimensões de um bloco sejam aumentadas, o procedimento adotado é dado por:

$$\begin{aligned}
 N_0 &\rightarrow N, \quad N > N_0 \\
 m_{n,k}^0 &= \left\lfloor \frac{n \cdot (N_0 - 1) + k}{N} \right\rfloor, \\
 m_{n,k}^1 &= \begin{cases} m_{n,k}^0 + 1, & m_{n,k}^0 < N_0 - 1, \\ m_{n,k}^0, & m_{n,k}^0 = N_0 - 1, \end{cases} \\
 \alpha_{n,k} &= n \cdot (N_0 - 1) - N \cdot m_{n,k}^0, \\
 V_n^s &= \left\lfloor \frac{\alpha_{n,k} \cdot (V_{m_{n,k}^1} - V_{m_{n,k}^0})}{N} \right\rfloor + V_{m_{n,k}^0}, \\
 n &= 0, 1, \dots, N - 1,
 \end{aligned} \tag{1}$$

onde  $V_n$  é o vetor original e  $V_n^s$  o escalonado.

Para que as dimensões do bloco sejam diminuídas:

$$\begin{aligned}
 N_0 &\rightarrow N, \quad N < N_0 \\
 m_{n,k}^0 &= \left\lfloor \frac{n \cdot (N_0 - 1) + k}{N} \right\rfloor, \\
 m_{n,k}^0 &= \begin{cases} m_{n,k}^0, & m_{n,k}^0 < N_0, \\ N_0 - 1, & m_{n,k}^0 = N_0, \end{cases} \\
 m_{n,k}^1 &= \begin{cases} m_{n,k}^0 + 1, & m_{n,k}^0 < N_0 - 1, \\ m_{n,k}^0, & m_{n,k}^0 = N_0 - 1, \end{cases} \\
 \alpha_{n,k} &= n \cdot (N_0 - 1) + k - N \cdot m_{n,k}^0, \\
 V_n^s &= \frac{1}{N_0 + 1} \sum_{k=0}^{N_0} \left( \left\lfloor \frac{\alpha_{n,k} \cdot (V_{m_{n,k}^1} - V_{m_{n,k}^0})}{N} \right\rfloor + V_{m_{n,k}^0} \right), \\
 n &= 0, 1, \dots, N - 1.
 \end{aligned} \tag{2}$$

A transformação bidimensional empregada neste trabalho pode ser obtida, para um bloco  $B^{m,n}$  sendo transformado em um bloco  $B^{k,l}$ , através do procedimento:

$$\begin{aligned}
 Y_i^{m,l} &= T_n^l [B_i^{m,n}], \quad i = 0, 1, \dots, m - 1, \\
 Z_j^{l,k} &= T_m^k \left[ \left( (Y^{m,l})^T \right)_j \right], \quad j = 0, 1, \dots, l - 1, \\
 B_h^{k,l} &= \left( (Z^{l,k})^T \right)_h, \quad h = 0, 1, \dots, k - 1,
 \end{aligned} \tag{3}$$

onde  $B_i^{m,n}$  representa a linha  $i$  do bloco  $B$  com dimensões  $m \times n$ .

No MMP bidimensional, um bloco de entrada  $\mathbf{X}$  é dividido em  $L$  blocos de mesmas dimensões que não se sobrepõem, sendo cada um representado por uma versão transformada  $\mathbf{S}_i^s$  de um elemento  $\mathbf{S}_i$  presente em um dicionário  $\mathcal{D}$ . Este processo está exemplificado na Figura 1(a).

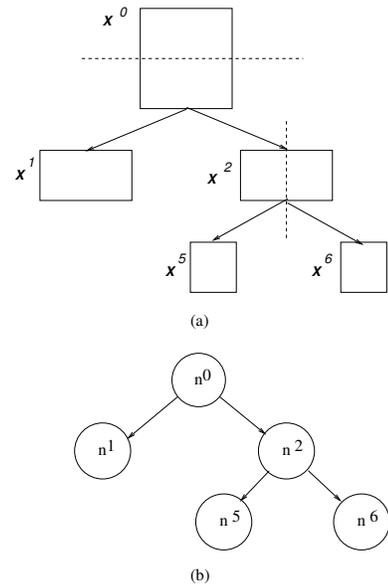


Fig. 1. Divisão de blocos no MMP.

A segmentação do bloco de entrada  $\mathbf{X}$  é representada por uma árvore de segmentação  $\mathcal{S}$ , conforme ilustrado na Figura 1(b). Cada nó  $n_j$  de  $\mathcal{S}$  está associado a um bloco  $\mathbf{X}^j$  do sinal, de tamanho  $(2^{-\lfloor \frac{p+1}{2} \rfloor} N \times 2^{-\lfloor \frac{p}{2} \rfloor} N)$ , onde  $p$  é a profundidade do nó  $n_j$  na árvore de segmentação  $\mathcal{S}$  e  $\lfloor x \rfloor$  é o maior inteiro menor ou igual a  $x$ . Um nó  $n_j$  da árvore de segmentação pode ter ou não filhos, dependendo do estado atual da mesma. No caso de tais filhos existirem, suas representações são  $n_{2j+1}$  e  $n_{2j+2}$ . Um nó que não tem filhos é chamado de nó-folha e consiste numa codificação efetiva do sinal, ou seja, apenas os nós-folhas da árvore de segmentação estão associados a elementos no dicionário, que são utilizados para aproximar cada um dos blocos de entrada.

O dicionário do MMP é atualizado da seguinte maneira: sempre que as aproximações  $\hat{\mathbf{X}}^{2j+1}$  e  $\hat{\mathbf{X}}^{2j+2}$  associadas aos nós-filhos  $n_{2j+1}$  e  $n_{2j+2}$  estão disponíveis, o MMP forma um elemento aproximado  $\hat{\mathbf{X}}^j$  para o bloco associado ao nó pai  $n_j$ , utilizando a concatenação das aproximações dos blocos associados aos dois nós-filhos. No exemplo da Figura 1, quando  $\hat{\mathbf{X}}^5$  e  $\hat{\mathbf{X}}^6$  estiverem disponíveis, é possível concatená-los e então obter uma nova aproximação  $\hat{\mathbf{X}}^2$ . Esta nova aproximação pode ser incluída no dicionário e reutilizada no futuro, caso algum bloco seja semelhante ao que ocasionou a inclusão. Este procedimento pode ser executado no codificador e repetido no decodificador, sem necessidade de se acrescentar *overhead* no arquivo. As únicas informações das quais o decodificador realmente necessita para realizar corretamente a reconstrução da imagem são aquelas relativas à árvore de segmentação e aos índices do dicionário correspondentes às aproximações dos nós-folhas. Deste modo, o algoritmo envia uma seqüência inteira  $i_m$ , consistindo nos índices do dicionário para a codificação de cada nó-folha, e uma outra seqüência de flags binários  $b_n$ , que especifica a árvore de segmentação  $\mathcal{S}$  para um dado bloco de imagem. A seqüência de flags representa a árvore de segmentação  $\mathcal{S}$  como uma série de decisões binárias, partindo-se do nó-raiz para cada nó-folha. Por exemplo, se o flag 0 fosse utilizado para indicar uma divisão de bloco, a árvore da Figura 1(b) seria representada por uma seqüência de flags 0,1,0,1,1.

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. O procedimento básico de otimização, como descrito em [1], começa com uma árvore de segmentação completa e segue dos nós-folhas para o nó-raiz, podando o par de nós-filhos  $n_{2j+1}$  e  $n_{2j+2}$  sempre que o custo Lagrangeano [11] da árvore que contém os mesmos for maior do que o custo da árvore de segmentação sem  $n_{2j+1}$  e  $n_{2j+2}$ . O custo Lagrangeano da árvore de segmentação  $\mathcal{S}$  é definido como  $J(\mathcal{S}) = D(\mathcal{S}) + \lambda R(\mathcal{S})$ , onde  $D(\mathcal{S})$  é a distorção obtida utilizando-se  $\mathcal{S}$  e  $R(\mathcal{S})$  é a taxa.

### III. O ALGORITMO SM-MMP

No MMP padrão, cada bloco de sinal é codificado de maneira independente, sem se levar em consideração as características dos seus vizinhos previamente codificados. Se a imagem processada não tiver muito conteúdo de alta frequência,

ou seja, se não possuir estruturas diversas alocadas por toda a sua extensão (muitas transições e bordas), é possível que o comportamento da vizinhança seja utilizado para se estimar o conteúdo de um dado bloco em codificação. O algoritmo proposto, chamado de SM-MMP, utiliza exatamente a informação adquirida dos blocos vizinhos que já foram codificados, com o objetivo de realizar uma predição sobre qual é o conjunto de elementos do dicionário mais provável para a codificação do bloco atual.

Para um dado bloco  $\mathbf{X}^j$  de dimensões  $n \times m$  associado ao nó  $n_j$  da árvore de segmentação  $\mathcal{S}$ , define-se o *vizinho superior*  $\mathbf{U}^j$  como o bloco *reconstruído* de dimensões  $n \times m$  imediatamente acima de  $\mathbf{X}^j$ , e o *vizinho esquerdo*  $\mathbf{L}^j$  como o bloco *reconstruído* de dimensões  $n \times m$  imediatamente à esquerda de  $\mathbf{X}^j$ . Esta notação está ilustrada na Figura 2. É importante observar que tanto  $\mathbf{U}^j$  quanto  $\mathbf{L}^j$  estão sempre disponíveis antes da codificação de  $\mathbf{X}^j$ , devido à ordem na qual o MMP processa os blocos de imagem de entrada e os elementos da árvore de segmentação.

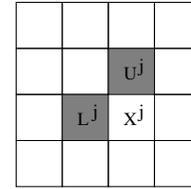


Fig. 2. Vizinhos superior e esquerdo.

Assim como no MMP padrão, utiliza-se um dicionário contendo blocos de dimensões  $n \times m$  para se codificar o bloco de entrada  $\mathbf{X}^j$ . Entretanto, no SM-MMP a probabilidade de ocorrência de cada elemento do dicionário é modificada de acordo com um *critério de continuidade*. No modelo adotado, a todos os elementos do dicionário que não obedecem ao critério de continuidade é atribuída probabilidade de ocorrência zero, o que é equivalente a reduzir o tamanho do dicionário original. O novo dicionário assim criado é chamado de *dicionário de estado*. Para construí-lo, a decisão de inclusão ou não de um dado elemento é baseada na medida de *rugosidade*  $r_{ij} = R(\mathbf{U}^j, \mathbf{L}^j, \mathbf{S}^i)$ , definida como:

$$R(\mathbf{U}^j, \mathbf{L}^j, \mathbf{S}^i) = \sum_{k=0}^{m-1} \left| \left| \frac{U_{m-2,k}^j - U_{m-1,k}^j + S_{0,k}^i - S_{1,k}^i}{2} \right| + S_{0,k}^i - U_{m-1,k}^j \right| + \sum_{p=0}^{n-1} \left| \left| \frac{L_{p,n-2}^j - L_{p,n-1}^j + S_{p,0}^i - S_{p,1}^i}{2} \right| + S_{p,0}^i - U_{p,n-1}^j \right| \quad (4)$$

Para codificar um dado bloco  $\mathbf{X}^j$ , o SM-MMP escolhe um dicionário de estado  $\mathcal{D}_s$  composto pelos  $N$  elementos  $\mathbf{S}_i$  do dicionário  $\mathcal{D}$  que possuem as menores medidas de rugosidade  $r_{ij}$ , de acordo com a equação (4). 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  $n \times m$  é avaliada através da função  $A(\mathbf{X})$ , definida como:

$$A(\mathbf{X}) = \max_{n,m} \left\{ \left( \sum_{k=0}^{m-1} |X_{n+1,k} - X_{n,k}| \right), \left( \sum_{p=0}^{n-1} |X_{p,m+1} - X_{p,m}| \right) \right\} \quad (5)$$

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

Para melhorar o desempenho do SM-MMP, modificou-se o procedimento de atualização do dicionário para que fossem incluídos deslocamentos dos blocos  $\tilde{\mathbf{X}}^j$  que já foram codificados, aumentando o tamanho do dicionário de um modo adaptativo e incluindo elementos com probabilidade razoável de serem utilizados. Esta técnica pode ser justificada pelo fato de que, numa imagem, pode haver estruturas semelhantes em diferentes posições (deslocadas). Um bom exemplo seria uma imagem com uma barra inclinada, como mostrado na Figura 3. Num determinado bloco, a borda da barra está um pouco deslocada em relação à do seu antecessor, e a inclusão de elementos deslocados para cima, no bloco antecessor, poderia facilitar em muito a sua codificação. Com isso, o dicionário  $\mathcal{D}$  cresce de modo mais acelerado, melhorando a codificação para os primeiros elementos e reduzindo a distorção média resultante das aproximações. É importante observar, ainda, que a taxa média não é afetada pelo crescimento mais acelerado do dicionário, haja vista que o tamanho  $N$  do dicionário de estado  $\mathcal{D}_s$  é independente do tamanho de  $\mathcal{D}$  (veja a equação (5) e a explicação que a sucede).

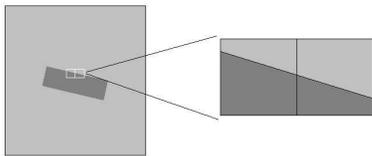


Fig. 3. Necessidade de inclusão de elementos deslocados.

Cada atualização de dicionário pode levar, agora, à inclusão de 10 elementos novos, a partir de deslocamentos na vertical, horizontal e diagonal, na direção dos blocos  $\tilde{\mathbf{X}}^j$  já codificados (deslocamentos em outras direções seriam incompletos ou necessitariam de repetição de pixels, ou seja, *stuffing*). Os deslocamentos podem ser de  $\frac{1}{4}$  do número de linhas,  $\frac{1}{4}$  do número de colunas ou  $\frac{1}{4}$  de cada (diagonal). Tal técnica, mostrada na Figura 4, leva a aumentos consideráveis na  $PSNR$  e na qualidade subjetiva das imagens codificadas.

#### IV. RESULTADOS DE SIMULAÇÕES

O MMP padrão e o SM-MMP com super-atualização de dicionário foram implementados em *linguagem C*, rodando em ambiente *Linux*. Os programas assim gerados foram utilizados para se comprimir imagens em níveis de cinza, inicialmente

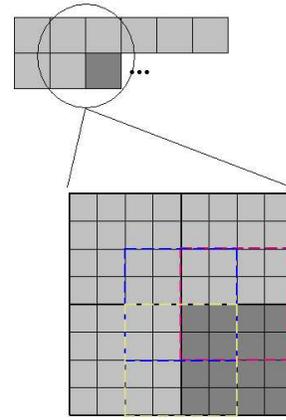


Fig. 4. Atualizações com deslocamentos de  $\frac{1}{2}m$  e  $\frac{1}{2}n$ .

divididas em blocos de  $16 \times 16$ . 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  $1 \times 1$  e foi iniciado com  $\mathcal{D}_0 = \{0, 2, \dots, 254\}$ . Os vetores em todas as outras escalas foram obtidos através de uma transformação de escala bidimensional *separável*, implementada utilizando-se operações de mudança de taxa de amostragem, como descrito em [1]. Com blocos iniciais de  $16 \times 16$ , 9 escalas diferentes podem ser obtidas (veja Figura 1(a)).

As Figuras 5, 6, 7, 8 e 9 mostram o desempenho taxa-distorção dos algoritmos para as imagens Lena, F-16, PP1205, PP1209 e Aerial, todas de  $512 \times 512$  pixels. Os resultados para o SMVQ (Side-Match Vector Quantizer) descrito em [3] (chamado no paper de GTSVQ e com resultados apenas para Lena e F-16) e para o SPIHT [5] também são apresentados, possibilitando uma comparação imediata entre os esquemas de compressão. Como os resultados para o JPEG2000 [4] são aproximadamente iguais aos do SPIHT, os mesmos não foram incluídos nos gráficos (na verdade, os resultados para o JPEG2000 tendem a ser levemente piores que os do SPIHT, pois devido à maior flexibilidade do primeiro, seus cabeçalhos acabam ficando maiores, o que diminui a eficiência do algoritmo). As imagens Lena, F-16 e Aerial foram obtidas no site <http://sipi.usc.edu/services/database/Database.html>. As imagens PP1205 e PP1209 foram digitalizadas do *IEEE Transactions on Image Processing*, volume 9, número 7, de julho de 2000. As páginas escolhidas foram as de número 1209 e 1205, que dão nome às imagens. A PP1205 contém apenas texto e fórmulas, ao passo que a PP1209 é 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 melhores desempenhos que o MMP para todas as imagens.
- 2) Tanto o SM-MMP quanto o MMP apresentaram melhores desempenhos que o SPIHT, com vantagens da ordem de  $\approx 2$ dB para a PP1209 (misto de imagens em níveis de cinza, texto e gráficos) e  $\approx 4$ dB para a PP1205 (texto e equações).
- 3) O SPIHT apresentou melhores desempenhos que o SM-

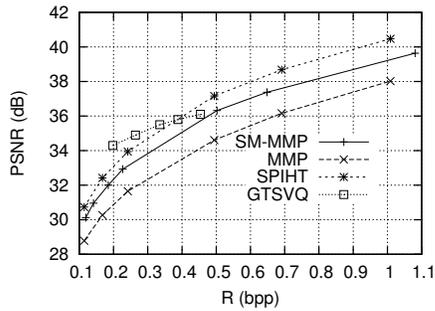


Fig. 5. Desempenho taxa-distorção para Lena 512 x 512.

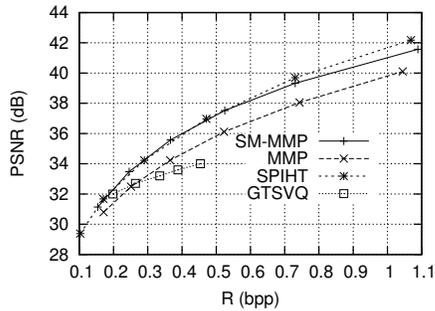


Fig. 6. Desempenho taxa-distorção para F-16 512 x 512.

MMP para a imagem Lena. O desempenho para as imagens F-16 e Aerial foi praticamente o mesmo para os dois algoritmos (para a imagem Aerial, o SM-MMP foi levemente superior).

- 4) O SM-MMP apresentou melhores desempenhos que o SMVQ [3] para a imagem F-16 em todas as taxas testadas. Para a imagem Lena, o SM-MMP apresentou resultados melhores em taxas acima de 0,4 bpp. Apesar da faixa apresentada na referência disponível [3] ser muito estreita, percebe-se uma clara tendência no aumento da vantagem proporcionada pelo SM-MMP com o aumento da taxa (não foi possível obter resultados para outras imagens comprimidas com o SMVQ ou para as imagens utilizadas neste trabalho em outras taxas).

A Figura 10 mostra a imagem Lena comprimida pelo SM-MMP e pelo MMP, a 0,3bpp. É fácil perceber a redução do efeito de blocagem, o que ocorre principalmente devido ao uso, por parte do SM-MMP, do critério de continuidade.

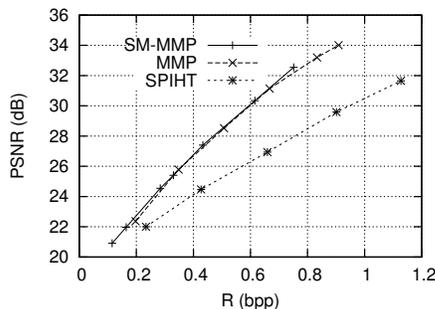


Fig. 7. Desempenho taxa-distorção para PP1205 512 x 512.

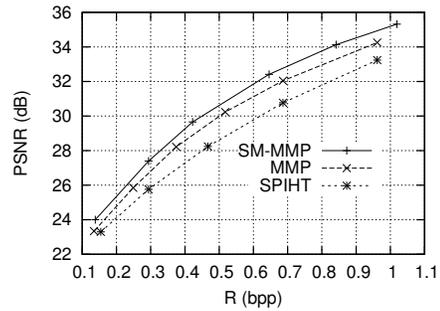


Fig. 8. Desempenho taxa-distorção para PP1209 512 x 512.

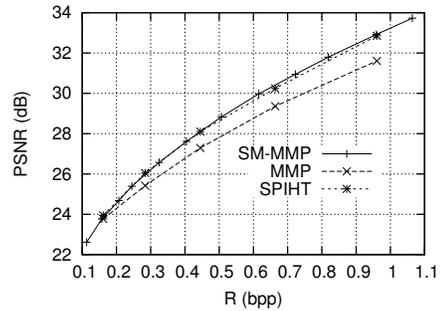


Fig. 9. Desempenho taxa-distorção para Aerial 512 x 512.

## V. CONSIDERAÇÕES FINAIS

Neste artigo, foi proposto um novo algoritmo chamado de SM-MMP, um esquema para compressão de sinais multidimensionais baseado no MMP. O algoritmo desenvolvido provou ser competitivo, alcançando ganhos significativos, no processamento de imagens suaves, com relação ao MMP padrão, através da introdução de um modelo estatístico para a fonte (imagem). Esta melhora foi obtida sem acarretar perda de desempenho na compressão de imagens contendo texto ou gráficos, nas quais o MMP padrão já apresentava resultados muito bons. O modelo adotado é extremamente simples e baseia-se nos métodos de SMVQ [2], [3], [7], [8], [9], [10] desenvolvidos para compressão de imagens. Espera-se que melhores resultados sejam obtidos com o aperfeiçoamento de tal modelo estatístico, utilizando-se, por exemplo, modelos de distribuição de Gibbs. Por último, vale a pena ressaltar a importância do desenvolvimento desta classe de algoritmos de compressão a qual pertence o SM-MMP, principalmente devido ao seu comportamento universal e ao potencial que apresenta para desenvolvimentos futuros.

## AGRADECIMENTOS

Este trabalho é resultado do programa de Mestrado interinstitucional UFAM-UFRJ/COPPE, financiado pela Superintendência da Zona Franca de Manaus-SUFRAMA. Os autores gostariam de agradecer à coordenadoria do programa de mestrado e à SUFRAMA pelo apoio recebido para a sua realização.

## REFERÊNCIAS

- [1] M. B. de Carvalho, E. A. B. da Silva and W. A. Finamore, "Multi-dimensional Signal Compression using Multiscale Recurrent Patterns,"



Fig. 10. Imagem Lena (detalhe): (a) MMP a 0.30 bpp. PSNR = 32.71 dB; (b) SM-MMP a 0.30 bpp. PSNR = 34.13 dB.

Elsevier's Signal Processing, Vol. 82, No. 11, pp. 1559-1580, November 2002.

- [2] 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.
- [3] 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.
- [4] D. S. Taubman, M. W. Marcellin, "JPEG2000: Image Compression Fundamentals, Standards, and Practice", Kluwer Academic Publishers, 2001.
- [5] A. Said and W.A. Pearlman, "A new, fast and efficient image codec based on set partitioning in hierarchical trees," IEEE Transactions on Circuits and Systems for Video Technology, vol.6, pp.243-250, June 1996.
- [6] W. B. Pennebaker and J. L. Mitchell, "JPEG Still Image Data Compression Standard", Van Nostrand Reinhold, 1994.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] 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.
- [11] M. M. Denn, "Optimization by Variational Methods", McGraw-Hill Book Company, 1969.