

COMPRESSÃO DE IMAGENS 3D DE RADAR METEOROLÓGICO USANDO RECORRÊNCIA DE PADRÕES MULTIESCALAS

Athayde L. V. Frauche¹, Murilo B. de Carvalho¹, Eduardo A. B. da Silva²

1. TET/CTC, Universidade Federal Fluminense. R. Passo da pátria, 156,
Niterói - RJ, 24210-240, BRAZIL

2. Universidade Federal do Rio de Janeiro, Cx. P. 68504,
Rio de Janeiro, RJ, 21941-972, BRAZIL

e-mails: frauche@hotmail.com, murilo@telecom.uff.br, eduardo@lps.ufrj.br

RESUMO

Neste trabalho, o algoritmo MMP ("Multidimensional Multiscale Parser") [1] é usado para comprimir imagens tridimensionais obtidas de radares meteorológicos. O MMP é baseado no casamento de padrões multiescalas. Ele codifica segmentos de um sinal de entrada usando versões contraídas e dilatadas de padrões armazenados em um dicionário. O dicionário é atualizado usando concatenações de segmentos previamente codificados. Sendo assim, o MMP constrói seu dicionário adaptativamente enquanto o dado é codificado. O Algoritmo pode ser adaptado para comprimir sinais com qualquer número de dimensões, e foi utilizado com sucesso na compressão de dados bidimensionais, como imagens, e unidimensionais, como sinais de ECG e EMG. Aqui, propomos uma versão tridimensional do algoritmo, que chamamos MMP-3D, e mostramos resultados de simulações onde o MMP-3D supera alguns dos melhores codificadores conhecidos.

Palavras-chave: casamento de padrões recorrentes, decomposição multiescalas, quantização vetorial, radar meteorológico

1. INTRODUÇÃO

O Radar Meteorológico Doppler, (DWR - "Doppler Weather Radar"), é uma ferramenta valiosa para geração de dados meteorológicos essenciais à hidrologia, hidrometeorologia, previsão do tempo, entre outras. Contudo, o grande volume de dados gerados pela rede de radares gera dificuldades para armazenamento, recuperação e transmissão em tempo real dos dados oriundos dos DWR. Por isso, um método de compressão de dados que consiga fornecer imagens de alta qualidade com altas taxas de compressão torna-se bastante útil. Em algumas aplicações, pode ser necessário manter os níveis de distorção tão baixos que a compressão sem perdas torna-se obrigatória. Em outras aplicações, como por exemplo na comunicação entre aeronaves e bases, a banda disponível é

limitada, demandando soluções que forneçam a maior qualidade possível a uma dada taxa. Neste cenário, o MMP surge como um bom candidato para compressão de dados DWR, uma vez que seu caráter universal provou ser valioso quando o algoritmo é empregado para comprimir os mais diversos tipos de fontes [2, 3]. Outra característica muito interessante é que o MMP é controlado por um único parâmetro que permite selecionar um ponto de operação com qualquer grau de distorção, inclusive zero.

A organização deste artigo é a seguinte: Na seção 2, O MMP original é descrito. Na seção 3, apresentamos a versão tridimensional do algoritmo, que chamamos MMP-3D. A seguir, na seção 4, mostramos os resultados experimentais obtidos bem como comparações com outros métodos de compressão de imagens e vídeo. Finalmente, na seção 5, apresentamos nossas conclusões.

2. O ALGORITMO MMP

O MMP é um método de compressão com perdas baseado no casamento de padrões multiescalas. Ele possui um dicionário $\mathcal{D} = \{\mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_{L-1}\}$ com L vetores \mathbf{v}_i de diferentes comprimentos que ele usa para codificar segmentos de diferentes comprimentos extraídos de um vetor de entrada $\mathbf{X}^0 = (x(0) \ x(1) \ \dots \ x(N-1))$, onde a dimensão N é uma potência de dois. Ao tentar codificar \mathbf{X}^0 , o MMP divide a sequência de entrada em vários segmentos disjuntos \mathbf{X}^j , e representa cada um por um elemento \mathbf{v}_{i_j} , pertencente ao dicionário \mathcal{D} , que irá ser usado para substituir \mathbf{X}^j . No Espírito do casamento de padrões multiescalas, qualquer vetor \mathbf{v}_i de \mathcal{D} pode ser usado para representar \mathbf{X}^j , não importando o seu comprimento. Para que isso seja possível, usa-se uma transformação de escala $T_{\ell(\mathbf{X}^j)}[\mathbf{v}_i]$, que ajusta o comprimento de \mathbf{v}_i de modo a igualar o de \mathbf{X}^j . A segmentação está associada a uma árvore binária de segmentação \mathcal{T} , onde cada nó n_j corresponde a um segmento \mathbf{X}^j , conforme ilustrado nas figuras 1 e 2.

Assim, a representação do vetor de entrada feita pelo

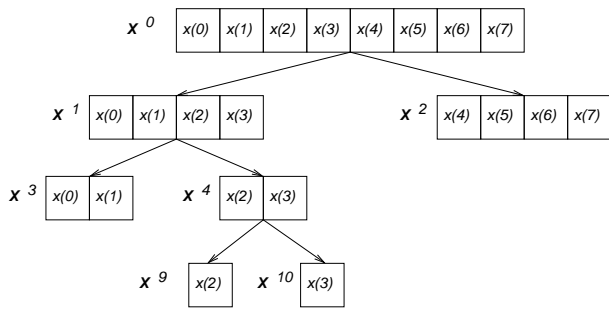


Fig. 1. A segmentation of the input vector \mathbf{X}^0

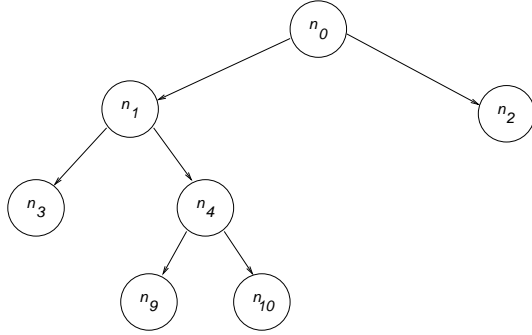


Fig. 2. A segmentation tree \mathcal{T}

MMP é composta pela árvore de segmentação binária \mathcal{T} e os índices i_j dos elementos \mathbf{v}_{i_j} do dicionário escolhidos para substituir cada segmento \mathbf{X}^j do vetor de entrada. Por exemplo, considerando-se a segmentação ilustrada na figura 1, a saída gerada pelo MMP seria a seqüência 0, 0, 1, i_3 , 0, 1, i_9 , 1, i_{10} , 1, i_2 . A seqüência de "flags" 0010111 define a árvore de segmentação escolhida a partir da raiz. Desde que os vetores associados a cada índice i_j estejam bem determinados (ou seja, tanto o codificador como o decodificador usem o mesmo dicionário \mathcal{D}), a seqüência gerada pelo MMP pode ser facilmente decodificada, do seguinte modo: o decodificador primeiro reconstrói a árvore de segmentação a partir dos "flags". Depois disso, o decodificador pode calcular os comprimentos dos vetores associados a cada índice i_j , uma vez que cada nodo n_j corresponde a um vetor de tamanho conhecido. Então, o decodificador substitui cada nodo n_j por uma versão adequadamente escalada do vetor \mathbf{v}_{i_j} pertencente ao dicionário \mathcal{D} .

A escolha dos melhores vetores e da melhor árvore de segmentação é baseada na minimização do custo Lagrangeano $J(\mathcal{T})$, definido por:

$$J(\mathcal{T}) = \lambda R_{\mathcal{T}} + \sum_{n_j \in \mathcal{T}_{\mathcal{L}}} (D(n_j) + \lambda R(n_j)) \quad (1)$$

onde $\mathcal{T}_{\mathcal{L}}$ é o conjunto de nodos folhas de \mathcal{T} , $D(n_j)$ é a distorção (por exemplo, o erro quadrático) obtida quando \mathbf{v}_{i_j} é usado para substituir \mathbf{X}^j , $R(n_j)$ é a taxa necessária para

codificar o índice i_j e $R_{\mathcal{T}}$ é a taxa necessária para codificar a árvore de segmentação.

Pode-se demonstrar que a árvore de segmentação que minimiza o custo Lagrangeano dado pela equação (1) também é ótima no sentido taxa-distorção [1].

O algoritmo MMP, conforme descrito até agora, opera com uma base de dados fixa, ou seja, o dicionário é estático. O desempenho melhora muito se permitimos que o dicionário se adapte ao dado sendo codificado. Isto é feito atualizando-se o dicionário com concatenações de segmentos anteriormente codificados, de um modo similar ao utilizado no algoritmo de compressão sem perdas de Lempel-Ziv [4]. Por exemplo, observando-se a figura 1, nota-se que assim que os índices i_9 e i_{10} são determinados, pode-se obter $\hat{\mathbf{X}}^9$ e $\hat{\mathbf{X}}^{10}$, que são as versões reconstruídas dos segmentos de entrada \mathbf{X}^9 e \mathbf{X}^{10} , pela aplicação de uma transformação de escalas apropriada aos vetores \mathbf{v}_{i_9} e $\mathbf{v}_{i_{10}}$. Em seguida obtemos a versão reconstruída de \mathbf{X}^4 concatenando $\hat{\mathbf{X}}^9$ e $\hat{\mathbf{X}}^{10}$, ou seja, $\hat{\mathbf{X}}^4 = (\hat{x}(2) \quad \hat{x}(3))$. Finalmente, podemos incluir $\hat{\mathbf{X}}^4$ no dicionário, uma vez que a informação utilizada para gerá-lo está disponível tanto no codificador quanto no decodificador, o que garante a sincronização dos dicionários de ambos. A próxima atualização do dicionário será feita, neste exemplo, pela inclusão da concatenação de $\hat{\mathbf{X}}^3$ e $\hat{\mathbf{X}}^4$, assim que os dois estejam disponíveis.

Durante a fase de otimização da árvore de segmentação, o MMP precisa comparar um segmento \mathbf{X}^j do vetor de entrada com todos os vetores do dicionário. Para isso ele tem que usar transformações de escala para ajustar o comprimento de cada vetor de modo a igualar o comprimento de \mathbf{X}^j . Contudo, se o comprimento do vetor de entrada é N , o procedimento de segmentação pode gerar apenas segmentos de comprimentos $N/2, N/4, \dots, 1$, ou seja o número de comprimentos distintos, ou escalas, é igual a $1 + \log_2(N)$. Para economizar tempo, podemos manter $1 + \log_2(N)$ cópias do dicionário, cada uma contendo versões escaladas para um mesmo comprimento de cada vetor do dicionário original. Assim, a transformação de escala só será utilizada no momento da inclusão de um novo vetor no dicionário. Se denotarmos por \mathcal{D}^p a cópia do dicionário na escala $2^{-p}N$, devemos incluir $T_{2^{-p}N}[\hat{\mathbf{X}}^j]$ em \mathcal{D}^p , $p = 0, 1, \dots, \log_2(N)$, sempre que formos incluir um novo elemento.

3. MMP TRIDIMENSIONAL (MMP-3D)

O procedimento de segmentação descrito na seção 2 pode ser estendido de muitas maneiras diferentes para processar dados de mais de uma dimensão. Em [1], uma extensão simples para o caso bidimensional é descrita, onde um bloco de entrada \mathbf{X}^0 de dimensão $N \times N$ pode ser subdividido em dois subblocos \mathbf{X}^1 e \mathbf{X}^2 de dimensão $N/2 \times N$. Cada subbloco pode ser independentemente subdividido em subsubblocos de dimensão $N/2 \times N/2$, \mathbf{X}^1 em \mathbf{X}^3 e \mathbf{X}^4 ; \mathbf{X}^2 em \mathbf{X}^5 e \mathbf{X}^6 , e assim por diante. Com esta regra, há $1 + 2 \log_2(N)$ escalas

possíveis: $N \times N$, $N/2 \times N$, $N/2 \times N/2$, $N/4 \times N/2$, ..., 1×1 . Esta regra simples poderia ser generalizada para o caso tridimensional, e no caso de blocos de entrada $N \times N \times N$ obteríamos $1 + 3 \log_2(N)$ escalas: $N \times N \times N$, $N/2 \times N \times N$, $N/2 \times N/2 \times N$, $N/2 \times N/2 \times N/2$, $N/4 \times N/2 \times N/2$, ..., $1 \times 1 \times 1$. Contudo, existem outras maneiras de subdividir os blocos. Por exemplo, poderíamos usar as seguintes escalas: $N \times N \times N$, $N \times N \times N/2$, $N \times N/2 \times N/2$, $N/2 \times N/2 \times N/2$, $N/2 \times N/2 \times N/4$, ..., $1 \times 1 \times 1$. Nós implementamos versões do MMP-3D usando as duas regras acima e observamos que o desempenho taxa-distorção varia muito, dependendo da imagem tridimensional particular testada e da regra escolhida. Concluímos que uma opção melhor é permitir que o MMP escolha, a cada etapa de segmentação, qual direção segmentar. Ou seja, para um bloco $N \times N \times N$, o MMP testa 3 possibilidades de segmentação: dois subbloco $N/2 \times N \times N$, ou dois sub-bloco $N \times N/2 \times N$, ou dois subbloco $N \times N \times N/2$, e escolhe a opção de melhor desempenho. Neste caso, a árvore de segmentação será descrita a partir da raiz por uma seqüência de "flags" quaternários: M, representando um casamento, A representando uma subdivisão na primeira direção, B representando uma subdivisão na segunda direção, e C representando uma subdivisão na terceira direção. Por exemplo, a figura 3 ilustra a segmentação de um bloco $4 \times 4 \times 4$ em um $4 \times 4 \times 2$ e mais dois $4 \times 2 \times 2$. Neste caso, a árvore é representada por: C, M, B, M, M .

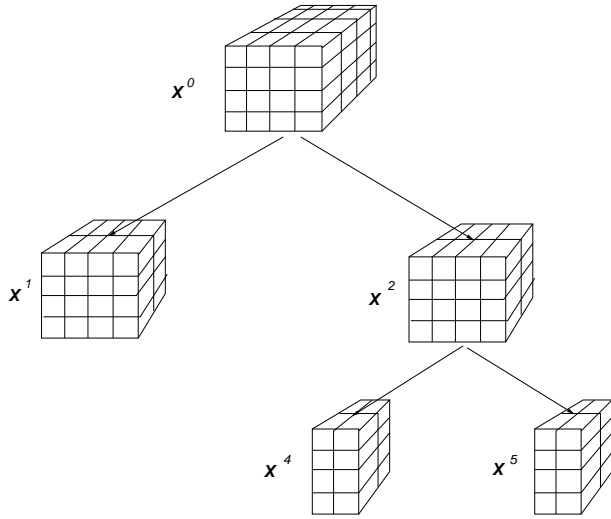


Fig. 3. Exemplo de segmentação tridimensional do bloco \mathbf{X}^0

A árvore de segmentação tridimensional otimizada pode ser obtida pelo seguinte procedimento recursivo:

$$(J, T_s) = \text{BestSeg}(\mathbf{X}^j, \mathcal{D}, \mathcal{T}, \lambda)$$

p1sso 1 Ache $\mathbf{v}_{i_j} \in \mathcal{D}$ tal que $J_0 = d(\mathbf{v}_{i_j}, \mathbf{X}^j) + \lambda R(i_j)$ seja mínimo, sendo $d(\mathbf{v}_{i_j}, \mathbf{X}^j)$ a distorção quando \mathbf{v}_{i_j} substitui \mathbf{X}^j e $R(i_j)$ a taxa necessária para codificar i_j .

- p2sso 2 Se a escala for $1 \times 1 \times 1$ retorne J_0 . Senão, vá para o passo 3.
- p3sso 3 Se n na escala corrente $n \times m \times p$ for maior do que um, subdivida \mathbf{X}^j em dois subbloco \mathbf{X}^{2j} e \mathbf{X}^{2j+1} de escala $n/2 \times m \times p$. Calcule $(J_1, T_1) = \text{BestSeg}(\mathbf{X}^{2j}, \mathcal{D}, \mathcal{T}, \lambda)$ e $(J_2, T_2) = \text{BestSeg}(\mathbf{X}^{2j+1}, \mathcal{D}, \mathcal{T}, \lambda)$.
- p4sso 4 Se m na escala corrente $n \times m \times p$ for maior do que um, subdivida \mathbf{X}^j em dois subbloco \mathbf{X}^{2j} e \mathbf{X}^{2j+1} de escala $n \times m/2 \times p$. Calcule $(J_3, T_3) = \text{BestSeg}(\mathbf{X}^{2j}, \mathcal{D}, \mathcal{T}, \lambda)$ e $(J_4, T_4) = \text{BestSeg}(\mathbf{X}^{2j+1}, \mathcal{D}, \mathcal{T}, \lambda)$.
- p5sso 5 Se p na escala corrente $n \times m \times p$ for maior do que um, subdivida \mathbf{X}^j em dois subbloco \mathbf{X}^{2j} e \mathbf{X}^{2j+1} de escala $n \times m \times p/2$. Calcule $(J_5, T_5) = \text{BestSeg}(\mathbf{X}^{2j}, \mathcal{D}, \mathcal{T}, \lambda)$ e $(J_6, T_6) = \text{BestSeg}(\mathbf{X}^{2j+1}, \mathcal{D}, \mathcal{T}, \lambda)$.
- p6sso 6 Calcule o custo total para codificar a sub-árvore de \mathcal{T} com raiz n_j : $J_M = J_0 + R_M$, $J_A = J_1 + J_2 + R_A$, $J_B = J_3 + J_4 + R_B$, $J_C = J_5 + J_6 + R_C$; onde R_M , R_A , R_B e R_C são as taxas necessárias para codificar os "flags".
- p7sso 7 Calcule $J_{ST} = \min(J_A, J_B, J_C)$. Se $J_A = J_{ST}$ faça $T_l = T_1$ e $T_r = T_2$. Se $J_B = J_{ST}$ faça $T_l = T_3$ e $T_r = T_4$. Se $J_C = J_{ST}$ faça $T_l = T_5$ e $T_r = T_6$.
- p8sso 8 Se $J_0 \leq J_{ST}$, retorne $J = J_0$. Senão, atualize \mathcal{T} incluindo T_l e T_r como as duas sub-árvores descendentes do nodo n_j , e então retorne $J = J_{ST}$.

A nova regra de segmentação produz um aumento significativo de desempenho taxa-distorção, mas a complexidade computacional também aumenta bastante. De fato, a regra simples, apresentada em [1], leva a $1 + 3 \log_2(N)$ escalas diferentes, quando estendida para o caso tridimensional com blocos de entrada $N \times N \times N$. A nova regra por outro lado, produzirá $(1 + \log_2(N))^3$ escalas diferentes, para o mesmo tamanho de bloco. Sendo assim, o número de operações de quantização vetorial (QV) que o MMP executa sobre o bloco, que são responsáveis pela maior parte da complexidade computacional [3], aumenta na mesma proporção.

4. RESULTADOS EXPERIMENTAIS

Nós implementamos o MMP-3D em software e usamos para comprimir dados tridimensionais obtidos de radares da rede de radares meteorológicos (WSR-88D) Norte-Americana NEXRAD. A imagem tridimensional de entrada foi inicialmente dividida em blocos de dimensão $8 \times 8 \times 4$ que foram processadas seqüencialmente pelo algoritmo. O dicionário

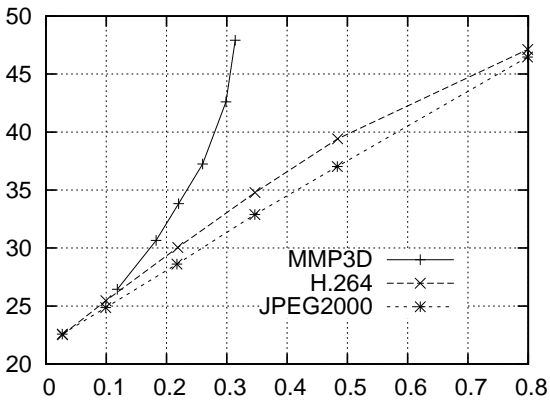


Fig. 4. Performance of 3D-MMP ($PSNR \times bits/sample$)

obtido após a codificação de um dado bloco foi usado como dicionário inicial para codificação do bloco seguinte. Para o primeiro bloco, usamos um dicionário inicial na escala $1 \times 1 \times 1$ composto de todos os inteiros no intervalo $[0, 255]$. Os dicionários iniciais nas demais escalas foram obtidos deste por meio de transformações de escala apropriadas.

A figura 4 mostra os resultados dos testes, $PSNR$ por taxa, do MMP-3D para uma imagem tridimensional de tamanho $512 \times 512 \times 4$ composta dos valores de refletividade obtidas das quatro elevações mais baixas do registro de número 6500KHGX20000610_000110 da base de dados NEXRAD. Na mesma figura, apresentamos os resultados obtidos com o codificador de vídeo H.264 [5], e com o codificador de imagens JPEG2000 [6]. Os resultados do H.264 foram obtidos com o software de referência JM.96, configurado para o melhor desempenho usando: GOP IPPP, otimização RD ligada, CABAC, taxa variável, formato YUV 4:0:0, janela de busca de 400 pixels, usando transformada de Haddamard. A imagem de entrada foi interpretada pelo H.264 como 4 quadros de 512×512 pixels cada, cada quadro correspondendo a uma elevação do radar. A figura 5 mostra os mesmos resultados, porém na forma de erro-quadrático por taxa.

Como pode ser observado, os resultados são muito bons, com o MMP superando o segundo colocado por 10 dB na taxa de 0.3 bits/amostra.

5. CONCLUSÕES

Nós aplicamos um método de compressão com perdas chamado MMP ao problema de codificação de imagens tridimensionais de radar. O MMP é baseado no casamento de padrões multiescalas, que é uma extensão do casamento de padrões convencional. O MMP usa um dicionário de padrões adaptativo e um procedimento simples de segmentação. Nós propusemos uma nova regra de segmentação 3D para o MMP que obteve um desempenho muito bom para a codificação das

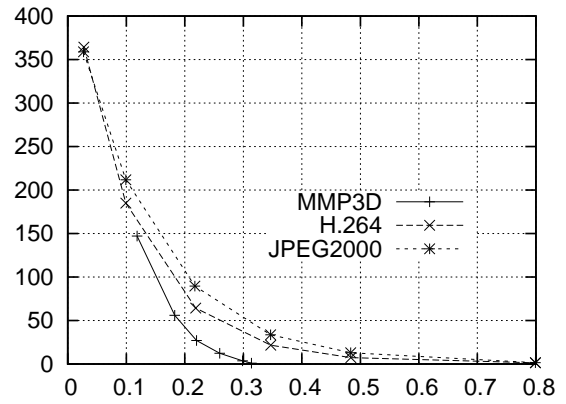


Fig. 5. Performance of 3D-MMP ($MSE \times bits/sample$)

imagens de radar. Embora nossa implementação do MMP seja de alta complexidade computacional, seu desempenho taxa-distorção é muito bom, superando alguns dos melhores codificadores conhecidos.

6. REFERENCES

- [1] M. de Carvalho, E. da Silva, and W. Finamore, "Multidimensional signal compression using multiscale recurrent patterns," *Elsevier Signal Processing*, no. 82, pp. 1559–1580, November 2002.
- [2] E. B. L. Filho, E. A. B. da Silva, M. B. de Carvalho, and F. S. Pinagé, "Universal image compression using multiscale recurrent patterns with adaptive probability model," to appear in *IEEE Transactions on Image Processing*, 2008.
- [3] E. B. L. Filho, E. A. B. da Silva, M. B. de Carvalho, W. S. S. Júnior, and J. Koiller, "Electrocardiographic signal compression using multiscale recurrent patterns," *IEEE Transactions on Circuits and Systems I*, vol. 52, no. 12, pp. 2739–2753, December 2005.
- [4] J. Ziv and A. Lempel, "Compression of individual sequences via variable-rate coding," *IEEE Transactions on Information Theory*, vol. 24, no. 5, pp. 530–536, September 1978.
- [5] Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6), *Draft of Version 4 of H.264/AVC (ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 part 10) Advanced Video Coding)*, March 2005.
- [6] D. S. Taubman and M. W. Marcelin, *JPEG2000: Image Compression Fundamentals, Standards and Practice*, Kluwer Academic Publishers, 2001.