

## 6 Referências

- [1] SLEPIAN, J.; WOLF, J. **Noiseless Coding of Correlated Information Sources**. In: IEEE Trans. on Information Theory, vol. 19, n 4, July 1973.
- [2] WYNER A.; ZIV, J. **The Rate-Distortion Function for Source Coding with Side Information at the Decoder**. In: IEEE Trans. on Information Theory, vol. 22, n1, January 1976.
- [3] PURI, R.; RAMCHANDRAN, K. **PRISM: A New Robust Video Coding Architecture Based on Distributed Compression Principles**. In: Proc. Allerton Conf., October 2002.
- [4] AARON, A.; ZHANG, R.; GIROD, B. **Wyner-Ziv Coding of Motion Video**. In: Proc. Asilomar Conference on Signals and Systems, Pacific Grove, CA, Nov. 2002. Invited Paper.
- [5] PURI, R.; RAMCHANDRAN, K. **Distributed source coding: Symmetric rates and applications to sensor networks**. In: Proc. IEEE Data Compression Conference, Snowbird, UT, Mar. 2000, pp. 363 –372.
- [6] PURI, R.; RAMCHANDRAN, K. **Group-theoretic construction and analysis of generalized coset codes for symmetric/asymmetric distributed source coding**. In: Proc. Conference on Information Sciences and Systems, Princeton, NJ, Mar. 2000.
- [7] PURI, R.; RAMCHANDRAN, K. **Geometric proof of rate-distortion function of Gaussian sources with side information at the decoder**. In: Proc. IEEE International Symposium on Information Theory (ISIT), Sorrento, Italy, June 2000, p. 351.
- [8] PRADHAN, S. S.; KUSUMA, J.; RAMCHANDRAN, K. **Distributed compression in a dense microsensor network**. In: IEEE Signal Processing Magazine, vol. 19, no. 2, Mar. 2002, pp. 51–60.
- [9] WANG, X.; ORCHARD, M. **Design of trellis codes for source coding with side information at the decoder**. In: Proc. IEEE Data Compression Conference, Snowbird, UT, Mar. 2001, pp. 361–370.

- [10] GARCÍA-FRÍAS J. **Compression of correlated binary sources using turbo codes.** In: IEEE Communications Letters, vol. 5, no. 10, pp. 417–419, Oct. 2001.
- [11] GARCÍA-FRÍAS J.; ZHAO, Y. **Data compression of unknown single and correlated binary sources using punctured turbo codes.** In: Proc. Allerton Conference on Communication, Control, and Computing, Monticello, IL, Oct. 2001.
- [12] BAJCSY, J.; MITRAN, P. **Coding for the Slepian-Wolf problem with turbo codes.** In: Proc. IEEE Global Communications Conference, vol. 2, San Antonio, TX, Nov. 2001, pp. 1400–1404.
- [13] MITRAN, P.; BAJCSY, J. **Near Shannon-limit coding for the Slepian-Wolf problem.** In: Proc. Biennial Symposium on Communications, Kingston, Ontario, June 2002.
- [14] AARON, A.; GIROD, B. **Compression with side information using turbo codes.** In: Proc. IEEE Data Compression Conference, Snowbird, UT, Apr. 2002, pp. 252–261.
- [15] ZHAO, Y.; GARCÍA-FRÍAS, J. **Joint estimation and data compression of correlated non-binary sources using punctured turbo codes.** In: Proc. Conference on Information Sciences and Systems, Princeton, NJ, Mar. 2002.
- [16] ZHAO, Y.; GARCÍA-FRÍAS, J. **Data compression of correlated non-binary sources using punctured turbo codes.** In: Proc. IEEE Data Compression Conference, Snowbird, UT, Apr. 2002, pp. 242–251.
- [17] MITRAN, P.; BAJCSY, J. **Coding for the Wyner-Ziv problem with turbolike codes.** In: Proc. IEEE International Symposium on Information Theory, Lausanne, Switzerland, June 2002, p. 91.
- [18] MITRAN, P.; BAJCSY, J. **Turbo source coding: a noise-robust approach to data compression.** In: Proc. IEEE Data Compression Conference, Snowbird, UT, Apr. 2002, p. 465.
- [19] ZHU, G.; ALAJAJI, F. **Turbo codes for non-uniform memoryless sources over noisy channels.** In: IEEE Communications Letters, vol. 6, no. 2, pp. 64–66, Feb. 2002.
- [20] GARCÍA-FRÍAS J.; ZHAO, Y. **Compression of binary memoryless sources using punctured turbo codes.** In: IEEE Communications Letters, vol. 6, no. 9, pp. 394–396, Sept. 2002.

- [21] GARCÍA-FRÍAS J.; ZHAO, Y. **Joint source-channel decoding of correlated sources over noisy channels.** In: Proc. IEEE Data Compression Conference, Snowbird, UT, Mar. 2001, pp. 283–292.
- [22] LIVERIS, A.; XIONG, Z.; GEORGHIADES, C. **Joint source-channel coding of binary sources with side information at the decoder using IRA codes.** In: Proc. Multimedia Signal Processing Workshop, St. Thomas, US Virgin Islands, Dec. 2002.
- [23] LIVERIS, A.; XIONG, Z.; GEORGHIADES, C. **Compression of binary sources with side information at the decoder using LDPC codes.** In: IEEE Communications Letters, vol. 6, no. 10, pp. 440–442, Oct. 2002.
- [24] LIVERIS, A.; XIONG, Z.; GEORGHIADES, C. **Compression of binary sources with side information at the decoder using LDPC codes.** In: Proc. IEEE Global Communications Symposium, Taipei, Taiwan, Nov. 2002.
- [25] SCHONBERG, D. S.; PRADHAN, S.; RAMCHANDRAN, K. **LDPC codes can approach the Slepian-Wolf bound for general binary sources.** In: Proc. Allerton Conference on Communication, Control, and Computing, Champaign, IL, Oct. 2002.
- [26] SCHONBERG, D. S.; PRADHAN, S.; RAMCHANDRAN, K. **Distributed code constructions for the entire Slepian-Wolf rate region for arbitrarily correlated sources.** In: Proc. IEEE Data Compression Conference, Snowbird, UT, Mar. 2004, pp. 292–301.
- [27] COLEMAN, T. et al. **On some new approaches to practical Slepian-Wolf compression inspired by channel coding.** In: Proc. IEEE Data Compression Conference, Snowbird, UT, Mar. 2004, pp. 282–291.
- [28] STANKOVIC, V. et al. **Design of Slepian-Wolf codes by channel code partitioning.** In: Proc. IEEE Data Compression Conference, Snowbird, UT, Mar. 2004, pp. 302–311.
- [29] LAN, C. F. et al. **Slepian-Wolf coding of multiple M-ary sources using LDPC codes.** In: Proc. IEEE Data Compression Conference, Snowbird, UT, Mar. 2004, pp. 549–549.
- [30] GEHRIG, N.; DRAGOTTI, P. L. **Symmetric and a-symmetric Slepian-Wolf codes with systematic and non-systematic linear codes.** In: IEEE Communications Letters, Jan. 2005, to appear.

- [31] WYNER, A. **On source coding with side information at the decoder.** In: IEEE Transactions on Information Theory, vol. IT-21, no. 3, May 1975, pp. 294–300.
- [32] SU, J. K.; EGGERS J. J.; GIROD, B. **Illustration of the duality between channel coding and rate distortion with side information.** In: Proc. 2000 Asilomar Conference on Signals and Systems, Pacific Grove, CA, Oct. 2000.
- [33] BARRON, R.; CHEN, J. B.; WORNELL, G. W. **The Duality Between Information Embedding and Source Coding with Side Information and Some Applications.** In: IEEE Transactions on Information Theory, vol. 49, no. 5, pp. 1159–1180, May 2003.
- [34] PRADHAN, S.; CHOU, J.; RAMCHANDRAN, K. **Duality between source coding and channel coding and its extension to the side information case.** In: IEEE Transactions on Information Theory, vol. 49, pp. 1181–1203, May 2003.
- [35] ZAMIR, R. **The rate loss in the Wyner-Ziv problem.** In: IEEE Transactions on Information Theory, vol. 42, no. 6, pp. 2073–2084, Nov. 1996.
- [36] ZAMIR, R.; SHAMAI. S. **Nested linear/lattice codes for Wyner-Ziv encoding.** In: Proc. Information Theory Workshop, Killarney, Ireland, June 1998, pp. 92–93.
- [37] ZAMIR, R.; SHAMAI. S.; EREZ, U. **Nested linear/lattice codes for structured multiterminal binning.** IEEE Transactions on Information Theory, vol. 48, no. 6, pp. 1250–1276, June 2002.
- [38] SERVETTO, S. D. **Lattice quantization with side information.** In: Proc. IEEE Data Compression Conference (DCC), Snowbird, UT, Mar. 2000, pp. 510–519.
- [39] XIONG, Z. et al. **Nested quantization and Slepian-Wolf coding: A Wyner-Ziv coding paradigm for i.i.d. sources.** In: Proc. IEEE Workshop on Statistical Signal Processing (SSP), St. Louis, MO, Sept. 2003.
- [40] LIU, Z et al. **Slepian-Wolf coded nested quantization (SWC-NQ) for Wyner-Ziv coding: Performance analysis and code design.** In: Proc. IEEE Data Compression Conference (DCC), Snowbird, UT, Mar. 2004, pp. 322–331.

- [41] YANG, Y. et al. **Wyner-Ziv coding based on TCQ and LDPC codes**. In: Proc. Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, Nov. 2003.
- [42] FLEMING, M.; ZHAO, Q.; EFFROS, M. **Network vector quantization**. In: IEEE Transactions on Information Theory, 2001, submitted.
- [43] LLOYD, S. P. **Least squares quantization in PCM**. In: IEEE Transactions on Information Theory, vol. IT-28, pp. 129–1373, Mar. 1982.
- [44] FLEMING M.; EFFROS, M. **Network vector quantization**. In: Proc. IEEE Data Compression Conference (DCC), Snowbird, UT, Mar. 2001, pp. 13–22.
- [45] MURESAN, D.; EFFROS, M. **Quantization as histogram segmentation: Globally optimal scalar quantizer design in network systems**. In: Proc. IEEE Data Compression Conference (DCC), Snowbird, UT, Apr. 2002, pp. 302–311.
- [46] EFFROS, M.; MURESAN, D. **Codecell contiguity in optimal fixed-rate and entropy-constrained network scalar quantizers**. In: Proc. IEEE Data Compression Conference (DCC), Snowbird, UT, Apr. 2002, pp. 312–321.
- [47] CARDINAL J.; ASCHE, G. V. **Joint entropy-constrained multiterminal quantization**. In: Proc. IEEE International Symposium on Information Theory (ISIT), Lausanne, Switzerland, June 2002, p. 63.
- [48] MONEDERO, D. R.; ZHANG, R.; GIROD, B. **Design of optimal quantizers for distributed source coding** In: Proc. IEEE Data Compression Conference (DCC), Snowbird, UT, Mar. 2003, pp. 13–22.
- [49] MONEDERO, D. R.; AARON, A.; GIROD, B. **Transforms for highrate distributed source coding**. In: Proc. Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, Nov. 2003.
- [50] WANG, Y.; OSTERMANN, ZHANG, Y. **Video Processing and Communications**, 1st ed. New Jersey: Prentice-Hall, 2002.
- [51] WIEGAND, T. et al. **Overview of the H.264/AVC video coding standard**. In: IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, no. 7, pp. 560–576, July 2003.
- [52] AARON, A. et al. **Wyner-Ziv coding for video: Applications to compression and error resilience**. In: Proc. IEEE Data Compression Conference, Snowbird, UT, Mar. 2003, pp. 93–102.

- [53] SETTON, E. et al. **Towards practical Wyner-Ziv coding of video**. In: Proc. IEEE International Conference on Image Processing, Barcelona, Spain, Sept. 2003.
- [54] RANE, S. et al. **Transform-domain Wyner-Ziv codec for video**. In: Proc. SPIE Visual Communications and Image Processing, San Jose, CA, Jan. 2004.
- [55] AARON, A.; RANE, S.; GIROD, B. **Wyner-Ziv video coding with hash-based motion compensation at the receiver**. In: Proc. IEEE International Conference on Image Processing, Singapore, Oct. 2004, to appear.
- [56] PURI, R.; RAMCHANDRAN, K. **PRISM: An uplink-friendly multimedia coding paradigm**. In: Proc. International Conference on Acoustics, Speech, and Signal Processing, Hong Kong, Apr. 2003.
- [57] PURI, R.; RAMCHANDRAN, K. **PRISM: A “reversed” multimedia coding paradigm**. In: Proc. IEEE International Conference on Image Processing, Barcelona, Spain, Sept. 2003.
- [58] ROWITCH, D.; MILSTEIN, L. **On the performance of hybrid FEC/ARQ systems using rate compatible punctured turbo codes**. In: IEEE Transactions on Communications, vol. 48, no. 6, pp. 948–959, June 2000.
- [59] ROWITCH, D.; MILSTEIN, L. **On the performance of hybrid FEC/ARQ systems using rate compatible punctured turbo codes**. In: IEEE Trans. on Communications, vol. 48, no. 6, pp. 948–959, June 2000.
- [60] PRADHAN, S. S.; RAMCHANDRAN, K. **Distributed source coding using syndromes (DISCUS): design and construction**. In: IEEE Trans. Inform. Theory 49 (3) (2003) 626–643.
- [61] HA, J.; MCLAUGHLIN, S. W. **Optimal puncturing of irregular low-density parity-check codes**. In: Proc. IEEE International Conf. on Communications, Anchorage, AK, 2003.
- [62] PISHRO-NIK, H.; FEKRI, F. **Results on punctured LDPC codes**. In: IEEE Inform. Theory Workshop, San Antonio, TX, 2004.
- [63] SESIA, S.; CAIRE, G.; VIVIER, G. **Incremental redundancy hybrid ARQ schemes based on low-density paritycheck codes**. In: IEEE Trans. Commun. 52 (8) (2004) 1311–1321.

- [64] CHEN, J. et al. **Distributed source coding using serially concatenated-accumulate codes**, In: IEEE Inform. Theory Workshop, San Antonio, TX, 2004.
- [65] JAGMOHAN, A.; SEHGAL, A.; AHUJA, N. **Predictive Encoding Using Coset Codes**. In: Proceedings of IEEE International Conference on Image Processing, Vol. 2, pp. 29-32, Rochester, New York, USA, September 2002.
- [66] KSCHISCHANG, F. R.; FREY, B. J.; LOELIGER, H. A. **Factor graphs and the sum-product algorithm**. In: IEEE Trans. Inform. Theory 47 (2) (2001), 498–519.
- [67] YANG, M.; RYAN, W. E.; LI, Y. **Design of efficiently encodable moderate-length high-rate irregular LDPC codes**. In: IEEE Trans. Commun. 52 (4) (2004), 564–571.
- [68] GIROD, B. et al. **Distributed Video Coding**. In: Proceedings of the IEEE, vol. 93, no. 1, January 2005.
- [69] BRITES, C.; ASCENSO, J.; PEREIRA, F. **Studying Temporal Correlation Noise Modeling for Pixel Based Wyner-Ziv Video Coding**. In: IEEE International Conference on Image Processing, Atlanta, USA, October 2006.
- [70] BRITES, C.; ASCENSO, J.; PEREIRA, F. **Content Adaptive Wyner-Ziv Video Coding Driven by Motion Activity**. In: IEEE International Conference on Image Processing, Atlanta, USA, October 2006.
- [71] VARODAYAN, D.; AARON, A.; GIROD, B. **Rate-Adaptive Codes for Distributed Source Coding**. EURASIP Signal Processing Journal, Special Section on Distributed Source Coding, vol. 86, no. 11, November 2006.
- [72] BRITES, C.; ASCENSO, J.; PEREIRA, F. **Improving Frame Interpolation with Spatial Motion Smoothing for Pixel Domain Distributed Video Coding**. In: 5th EURASIP Conference on Speech and Image Processing, Multimedia Communications and Services, Smolenice, Slovak Republic, July 2005.
- [73] KLOMP, S.; VATIS, Y.; OSTERMANN, J. **Side Information Interpolation with Sub-pel Motion Compensation for Wyner-Ziv Decoder**. In: Int. Conf. on Signal Processing and Multimedia Applications, Setúbal, Portugal, August 2006.
- [74] KUBASOV, D.; LAJNEF, K.; GUILLEMOT, C. **A Hybrid Encoder/Decoder Rate Control for Wyner-Ziv Video Coding with a**

- Feedback Channel.** In: IEEE Int. Workshop on Multimedia Signal Processing, Crete, Greece, October 2007.
- [75] KUBASOV, D.; GUILLEMOT, C.; NAYAK, J. **Optimal Reconstruction in Wyner-Ziv Video Coding with Multiple Side Information.** In: IEEE Int. Workshop on Multimedia Signal Processing, Crete, Greece, October 2007.
- [76] LUBY, M. **LT Codes.** In: Proceedings of the 43rd Annual IEEE Symposium on the Foundations of Computer Science (STOC), 2002, pp.271-280.
- [77] MACWILLIAMS, F. J.; SLOANE, N. J. A. **The Theory of Error Correcting Codes.** Elsevier-North-Holland, 1977.
- [78] FORNEY, G. D. **Coset Codes-Part I: Introduction and Geometrical Classification.** In: IEEE Transactions on Information Theory, vol. 3*N*, pp. 1123–1151, September 1988.
- [79] FORNEY, G. D. **Coset Codes-Part II: Binary Lattices and Related Codes.** In: IEEE Transactions on Information Theory, vol. 34, pp. 1152–1187, September 1988.
- [80] CHUNG, S. Y.; RICHARDSON, T. J.; URBANKE, R. L. **Analysis of sum-product decoding of low-density parity check codes using a Gaussian approximation.** In: IEEE Trans. Inform. Theory 47 (2), (2001) 657–670.
- [81] RICHARDSON, R. L. et al. **Finite-length analysis of low-density parity-check codes on the binary erasure channel.** In: IEEE Trans. Inform. Theory 48 (6) (2002) 1570–1579.
- [82] VILLASENOR, R. D. et al. **Construction of irregular LDPC codes with low error floors.** In: Proc. IEEE International Conf. on Communications, Anchorage, AK, 2003.
- [83] RAMAMOORTHY, A. R.; WESEL, D. **Construction of short block length irregular low-density parity-check codes,** In: Proc. IEEE International Conf. on Communications, Paris, France, 2004.
- [84] ISO/IEC International Standard 14496-10:2003. **Information Technology – Coding of Audio-visual Objects – Part 10: Advanced Video Coding.**
- [85] CLARKE, R. **Transform Coding of Images.** In: Academic Press, San Diego, USA, 1990.
- [86] RAO, K. R.; YIP, P. **Discrete Cosine Transform: Algorithms, Advantages, Applications.** Academic Press, Boston, 1990.



- [87] HASKELL, B.; PURI, A.; NETRAVALI, A. **Digital Video: An Introduction to MPEG-2**. Chapman & Hall, New York, USA, 1997.
- [88] ALPARONE, L. et al. **Adaptively Weighted Vector-Median Filters for Motion-Fields Smoothing**. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 4, pp. 2267-2270, Georgia, USA, May 1996.
- [89] GALLAGER, R. G. **Low-density parity-check codes**. In: IRE Transactions on Information Theory, vol. IT-8, pp. 21-28, Janeiro de 1962.
- [90] WIBERG, N.; LOELIGER, H.; KÖTTER, E. R. **Codes and iterative decoding on general graphs**. In: European Transactions on Telecommunications, vol. 6, pp. 513-525, Setembro/Outubro de 1995.
- [91] KSCHISCHANG, F. R.; FREY, B. J.; LOELIGER, E. H. **Factor graphs and the sum-product algorithm**. In: IEEE Transactions on Information Theory, Vol. 47, n° 2, pp. 498-519, Fevereiro de 2001.
- [92] PEARL, J. **Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference**. Morgan Kaufmann Publishers, Inc., 1988.
- [93] JELINEK, F. et al. **Optimal decoding of linear codes for minimizing symbol error rate**. In: IEEE Trans. on Information Theory, pp. 284-287, Março de 1974.
- [94] BARJATYA, A. **Block Matching Algorithms For Motion Estimation**. DIP 6620, Final Project Paper, Spring 2004.

## 7 Apêndice A – Construção do Código LDPC

Os códigos LDPC são especificados por uma matriz de verificação de paridade indicada por  $H$ , a qual é esparsa, ou seja, consiste principalmente em 0s e poucos 1s. Em especial, falamos de códigos LDPC  $(n, t_c, t_r)$ , onde  $n$  indica o tamanho de bloco;  $t_c$ , o peso (i. e., o número de 1s) em cada coluna da matriz  $H$  e  $t_r$ , o peso de cada linha, com  $t_r > t_c$ . A taxa desse código LDPC é

$$r = 1 - \frac{t_c}{t_r} \quad (7.1)$$

cuja validade pode ser justificada da seguinte maneira. Admitamos que  $\rho$  indique a densidade de 1s na matriz  $H$  de verificação da paridade. Então, chamando de  $\lambda$  o comprimento da palavra-código e  $n$  o número de bits da mensagem, pode-se definir

$$t_c = \rho(n - \lambda) \quad (7.2)$$

$$e \quad t_r = \rho n \quad (7.3)$$

onde  $(n - \lambda)$  é o número de linhas em  $\mathbf{A}$  e  $n$ , o número de colunas (i.e., o tamanho de bloco). Portanto, dividindo  $t_c$  por  $t_r$ , obtemos

$$\frac{t_c}{t_r} = 1 - \frac{\lambda}{n} \quad (7.4)$$

Por definição, a taxa de código de um código de blocos é  $\lambda/n$ , por isso o resultado da eq. (7.1). Entretanto, para que esse resultado persista, as linhas de  $\mathbf{A}$  devem ser linearmente independentes.

A estrutura dos códigos LDPC é bem retratada por grafos bipartites. Por exemplo, em um código LDPC com  $n = 10$ ,  $t_c = 3$  e  $t_r = 5$ , os nós de um lado, por exemplo da esquerda do grafo, são nós variáveis, os quais correspondem a elementos da palavra-código, enquanto que os nós da direita (por exemplo), são nós de verificação, os quais correspondem ao conjunto de restrições de verificação da paridade satisfeitas pelas palavras-código do código. Diz-se que os códigos

LDPC deste tipo são regulares porque todos os nós de um tipo similar têm exatamente o mesmo grau. Neste exemplo, o grau dos nós variáveis é  $t_c = 3$ , e o grau dos nós de verificação é  $t_r = 5$ . À medida que o tamanho de bloco  $n$  se aproxima do infinito, cada nó de verificação é conectado a uma fração cada vez menor de nós variáveis; daí, o termo baixa densidade.

A matriz  $H$  é construída colocando-se aleatoriamente 1s em  $H$ , o que está sujeito a restrições de regularidade.

- Cada coluna contém um número fixo pequeno,  $t_c$ , de 1s.
- Cada linha contém um número fixo pequeno,  $t_r$ , de 1s.

Na prática, essas restrições de regularidade muitas vezes são ligeiramente quebradas, a fim de evitar ter linhas linearmente dependentes na matriz  $H$  de verificação da paridade.

Para propósitos de codificação, podemos obter uma matriz geradora  $G$  para códigos LDPC por meio de eliminação gaussiana executada em aritmética módulo 2; esse procedimento será ilustrado posteriormente no Exemplo 7.1 mais adiante. O vetor código  $c$  1 por  $n$  é particionado primeiramente como

$$c = [b:m] \quad (7.5)$$

onde  $m$  é o vetor mensagem  $\lambda$  por 1 e  $b$  o vetor paridade  $(n - \lambda)$  por 1. De maneira semelhante a matriz  $H$  de verificação da paridade é particionada como

$$H^T = \begin{bmatrix} H_1 \\ \dots \\ H_2 \end{bmatrix} \quad (7.6)$$

onde  $H_1$  é uma matriz quadrada de dimensões  $(n - \lambda) \times (n - \lambda)$ , e  $H_2$ , uma matriz retangular de dimensões  $\lambda \times (n - \lambda)$ ; a transposição simbolizada pelo sobrescrito  $T$  é usada no particionamento da matriz  $H$  para facilitar a apresentação. Impondo a restrição ao código LDPC, podemos escrever

$$[b:m] \begin{bmatrix} H_1 \\ \dots \\ H_2 \end{bmatrix} = 0 \quad (7.7)$$

ou então,

$$bH_1 + mH_2 = 0 \quad (7.8)$$

os vetores  $m$  e  $b$  se relacionam por

$$b = mP \quad (7.9)$$

onde  $P$  é a matriz coeficiente. Então, substituindo essa relação na eq. (7.8), descobrimos facilmente que, para qualquer vetor mensagem  $m$  diferente de zero, a matriz coeficiente de códigos LDPC satisfaz à condição

$$PH_1 + H_2 = 0 \quad (7.10)$$

a qual persiste para todos os vetores-mensagem diferentes de zero e, em especial, para  $m$  na forma  $[0\dots 0 \ 1 \ 0\dots 0]$ , que isolam linhas individuais da matriz geradora.

Resolvendo essa equação para a matriz  $P$ , obtemos

$$P = H_2 H_1^{-1} \quad (7.11)$$

onde  $H_1^{-1}$  é a matriz inversa de  $H_1$ , a qual é naturalmente definida na aritmética módulo 2. Finalmente, a matriz geradora de códigos LDPC é definida por

$$\begin{aligned} G &= [P : I_\lambda] \\ &= [H_2 H_1^{-1} : I_\lambda] \end{aligned} \quad (7.12)$$

onde  $I_\lambda$  é uma matriz identidade de ordem  $\lambda$ .

Exemplo 7.1: construção da matriz geradora do Código LDPC (10, 3, 5).

Considere um grafo bipartite pertinente a um código LDPC (10, 3, 5). A matriz de verificação de paridade do código é definida por

$$H = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 1 & : & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & : & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & : & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & : & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & : & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & : & 1 & 1 & 1 & 1 \end{bmatrix} \quad (7.13)$$

$\underbrace{\hspace{10em}}_{A_1^T} \quad \underbrace{\hspace{10em}}_{A_2^T}$

a qual parece ser aleatória, embora mantenha as restrições quanto à regularidade:

$t_c = 3$  e  $t_r = 5$ . Particionando a matriz  $H$  da maneira descrita na eq. (7.6), teremos:

$$H_1 = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \quad (7.14)$$

$$H_2 = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \quad (7.15)$$

O inverso da matriz  $H_1$  é, portanto,

$$H_1^{-1} = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 \end{bmatrix} \quad (7.16)$$

A matriz produto  $H_2 H_1^{-1}$  é (usando o valor dado de  $H_2$  e o valor de  $H_1^{-1}$  que acabamos de encontrar)

$$H_2 H_1^{-1} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix} \quad (7.17)$$

Finalmente, usando a eq. (7.12), chega-se a matriz geradora do código LDPC(10, 3, 5):

$$G = \left[ \begin{array}{cccccc|cccc} 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{array} \right] \quad (7.18)$$

$\underbrace{\hspace{10em}}_{H_2 H_1^{-1}} \quad \underbrace{\hspace{10em}}_{I_\lambda}$

É importante reconhecer que o código LDPC descrito neste exemplo se destina somente a ilustrar o procedimento envolvido na geração desse código. Na prática, o tamanho de bloco  $n$  é muito maior do que o considerado neste exemplo. Além disso, ao construir a matriz  $H$ , podemos restringir todos os pares de colunas para que tenham uma sobreposição matricial (isto é, o produto interno de duas colunas quaisquer da matriz  $H$ ) que não ultrapasse 1; espera-se que essa restrição, antes e acima das restrições quanto à regularidade, melhore o desempenho dos códigos LDPC.

## 8 Apêndice B – Opções de Configuração do Codec Proposto

```
//===== Dados de Entrada =====
row = 144 //QCIF: row=144, col=176 (default)
col = 176 //SCIF: row=128, col=96; CIF: row=352, col=288
N = 4; //DCT NxN
TQ = 7; //valor de 1 a 18 (18 => melhor qualidade)
passo_estimacao = 1; //(1 a 14) = melhor (mais demorado) a
//pior (mais rápido)
num_iteracoes = 50;
err = 0.001; //percentual de erro admissível
QP_intra = 32;
CN = 1; //(0: normalização não habilitada; 1: habilitada;
//3: habilitada, com coef_Norm = valor máximo de WZ)

Coef_Norm = 255; //(240 a 255)
exclusoes = 0; //número de exclusões de bitplanes
//transmitidos (1 a 3)

BE = 1; //0: calcula os vetores de movimewnto utilizando o
//método de busca exaustiva
//1: métodos mais rápidos (DS para blocos 8x8 e ARPS
//para 16x16)

Met_Quant = 1; //Método de quantização. 0: envio da faixa
//dinâmica completa; 1: envio somente do
//limite superior;

Correl = 1; //Fórmula para cálculo da correlação. 1: MAD
//2: SAD; 3: MSE

//=== Qualificação da Sequência quanto ao Movimento =====
qtde_movim = 3; //quantidade de movimento
//muito lento = 1 (consultório, escritório, portão de casa)
//lento = 2 (loja)
//médio = 3 (banco)
//rápido = 4 (pedestre na calçada)
//muito rápido = 5 (estação de trem)

//=== Tipo de Procedimento para Cálculo do ALFA =====
proceeding = 3; //procedimento
//nível de frame = 1
//nível de banda = 2
//nível de bloco = 3
//nível de coeficiente = 4
```

## 9

## Apêndice C – Algoritmo Proposto para Decodificação dos Bitplanes

```

// Linguagem de programação: Matlab
// === inicialização de variáveis =====
num_blocos = row*col/N^2;
Lmax = zeros(N^2-1, num_blocos);
Lmin = zeros(N^2-1, num_blocos);
//Lmax e Lmin: limite superior e inferior de cada banda
bitplane_rec = zeros(size(bitplanes));
duracao = zeros(1,N^2-1);

// === decodificação dos bitplanes =====
Para1 ban=1:N^2-1 //ban = banda = k
Tic //tic = marcador de tempo
fprintf('Banda %d \n', ban)
p1 = zeros(1,num_blocos); //probabilidade a posteriori
    //p1(j)=P(Bit(m,j) = 1|Y(m,j),Bit(m+1,j))

Lmax(ban,:) = Vmax(ban)*ones(1,num_blocos);

Se2 ban~=1
    Lmin(ban,:) = -Lmax(ban,:);
Fim2

Para3 bitp = num_bits(ban):-1:1 //bitp = bitplane atual
    //bitplane(:, :, num_bits)=> MSB
    //bitplane(:, :, 1) => LSB

resp=1;
Enquanto4 resp == 0 (enquanto a decodif. for incompleta)
    fprintf('Decodificando Bitplane %d \n', bitp)
    bitplane_rec(ban,:,bitp) = bitplanes_Y(ban,:,bitp);

    //==== Cálculo da P(Bitp_m-1,j = 1 / Y, Btp_m,j) =====
    //===== e dos bitplanes reconstruídos =====
    p0 = zeros(N^2-1,num_blocos,max(num_bits));
    p1 = zeros(N^2-1,num_blocos,max(num_bits));

Para5 j=1:num_blocos
    //fprintf('Decodificação do coef. %d \n', j)
    Switch6 proceeding
        caso 1 //nível de frame
            F=@(x)alfa/2*exp(-alfa*abs(Y_banda(ban,j)-x))
        caso 2 //nível de banda
            F = @(x)alfa(ban)/2*
                *exp(-alfa(ban)*abs(Y_banda(ban,j)-x))
        caso 3 //nível de bloco
            F = @(x) alfa_bloco(ban,j)/2 *

```



```

        exp(-alfa_bloco(ban,j)*abs(Y_banda(ban,j)-x))
    caso 4 //nível de coeficiente
        F = @(x) alfa_pixel(ban,j)/2*
            exp(-alfa_coef(ban,j)*abs(Y_banda(ban,j)-x))
    Fim6
    area_0=quad(F,Lmin(ban,j),(Lmax(ban,j)+Lmin(ban,j))/2)
    area_1=quad(F,(Lmax(ban,j)+Lmin(ban,j))/2,Lmax(ban,j))
    p1(j) = area_1/(area_0 + area_1);
    //p0(j) = 1-p1(j);
    Fim5

    [bitp_rec,resp]=decod_LDPC(p1,sind,num_iteracoes,err);
        //sind = síndromes acumuladas
    Para7 j=1:num_blocos
        Se8 bitp_rec(j) == 0 //se o bit j foi estimado como 1
            Lmax(ban,j)=(Lmax(ban,j)+Lmin(ban,j))/2;
        Caso contrário:
            Lmin(ban,j)=(Lmax(ban,j)+Lmin(ban,j))/2;
        Fim8
    Fim7
        bitplane_rec(ban,:,bitp) = bitp_rec;
    Se9 resp=1 (decodificação mal sucedida)
        sind = req_mais_sind();
    Fim9
    Fim4
    Fim3
    duracao(ban) = toc; //duração da codificação da banda
    Fim1

```