Lossless Bit-plane Compression of Microarray Images Using 3D Context Models*

António J. R. Neves, Armando J. Pinho, António R. C. Paiva†
Dept. de Electrónica e Telecomunicações / IEETA
Universidade de Aveiro
3810–193 Aveiro, Portugal
an@ieeta.pt, ap@det.ua.pt, arpaiva@ufl.edu

ABSTRACT
With the recent growth of interest in microarray technology and the improvement of the technology responsible for producing these images, massive amounts of microarray images are currently being produced. In this paper, we present a new lossless method for efficiently compress microarray images based on arithmetic coding using a 3D context model. Our method produces an embedded bitstream that allows progressive decoding. We present the compression results using 49 publicly available images and we compare these results with three image coding standards: lossless JPEG2000, JBIG and JPEG-LS. The proposed method gives better results for all images of the test set.

KEY WORDS
Microarray images; lossless image compression; image coding standards; bit-plane coding; context models; lossy to lossless.

1 Introduction

The DNA microarray technology is a new and effective tool for biomedical research. It allows the processing of thousands of genes simultaneously [1]. The result of a microarray experiment is typically two images of 16 bits per pixel, obtained after scanning the microarray slide with a laser and capturing the light emitted by two different fluorescent markers. Usually, a green marker (Cy3) is used to label the reference sample, whereas a red marker (Cy5) labels the sample under analysis. Depending on the size of the array and the resolution of the scanner, these images may require from a few megabytes to several tens of megabytes of storage.

The number of microarray experiments is increasing and, due to the huge amount of space needed for storing each image and the need for efficient transmission, finding good techniques to compress microarray images is an important challenge. The motivation for developing lossless methods is because the existing analytic methods that are used to process the images are in permanent development, being imprudent, at least nowadays, discarding the raw data and keeping only the parameters obtained through analysis. In other words, it seems wise to keep the microarray images free of losses, in order to facilitate future re-analysis by better algorithms. The cost of these experiments might also discourage experiment repetition. Finally, we should note that methods that are progressive, allowing lossy to lossless decoding, are of high interest, specially in the case where remote databases have to be accessed using transmission channels of reduced bandwidth.

In this paper, we present a compression method based on arithmetic coding driven by a 3D context model. This method was inspired by the work of Yoo *et al.*, named EIDAC (Embedded Image Domain Adaptive compression of Simple Images), regarding the compression of simples images, i.e., images with only a reduced number of intensities [22]. The image is compressed on a bit-plane basis, going from the most significant bit-plane to the least significant bit-plane. Encoding is stopped if an average of more than 1 bit/pixel is obtained after encoding a given bit-plane. In this case, the remainder bit-planes are sent uncompressed. The context model used by the arithmetic encoder uses (causal) pixels from the bit-plane under compression and also pixels from the bit-planes already encoded. The results obtained are compared with three standard image coding techniques, namely, lossless JPEG2000 [2–4], JBIG [5, 6] and JPEG-LS [4, 7, 8].

Some techniques have already been proposed for microarray image compression, as we show in the next Section. However, it is often difficult to compare the performance among them and / or in relation to current image compression standards or with new methods under development. One of the factors that contributes to this limitation is the lack of results on a common and representative set of images. In this work, we tried to overcome this drawback, by providing compression results on a set of 49 images gathered from three different publicly available sources.

In the remainder of this paper, we give a brief overview of the microarray image compression methods that have been proposed in the literature. Then, we briefly introduce the three image coding standards addressed in

*This work was supported in part by the Fundação para a Ciência e a Tecnologia (FCT) and by the INFOBIOMED European NoE.

†The author is now with the University of Florida, USA.

this paper, namely, lossless JPEG2000, JBIG and JPEG-LS. Next, we explain our algorithm and present the compression results obtained. Finally, we draw some conclusions.

2 Image coding methods

2.1 Specialized methods

To the best of our knowledge, at the time of writing, there are four published methods for the lossy and / or lossless compression of microarray images, namely, the works of Jörnsten *et al.* [9–12], Hua *et al.* [13, 14], Faramarzpour *et al.* [15,16], and Lonardi *et al.* [17]. Although different, they share, nevertheless, a common structure that is depicted in Fig. 1. Next, we give a brief overview of each of these techniques.

The technique proposed by Jörnsten *et al.* [9–12] is characterized by a first stage devoted to gridding and segmentation. Using the approximate center of each spot, a seeded region growing is performed for segmenting the spots. The segmentation map is encoded using chain-coding, whereas the interior of the regions are encoded using a modified version of the LOCO-I (LOw COmplexity LOssless COmpression for Images) algorithm (this is the algorithm behind the JPEG-LS coding standard), named SLOCO. Besides lossy to lossless capability, Jörnsten's technique allows partial decoding, by means of independently encoded image blocks.

Hua *et al.* [13,14] presented a transform-based coding technique. Initially, a segmentation is performed using the Mann-Whitney algorithm, and the segmentation information is encoded separately. Due to the thresholding properties of the Mann-Whitney algorithm, the gridding stage depicted in Fig. 1 is avoided. Then, a modified EBCOT (Embedded Block Coding with Optimized Truncation) for handling arbitrarily shaped regions is used for encoding the spots and background separately, allowing lossy to lossless coding of background only (with the spots losslessly encoded) or both background and spots. EBCOT is the technique used by the JPEG2000 image coding standard for encoding the coefficients of the discrete wavelet transform [4].

The compression method proposed by Faramarzpour *et al.* [15, 16] starts by locating and extracting the microarray spots, isolating each spot into an individual ROI (region of interest). To each of these ROI's, a spiral path is adjusted such that its center coincides with the center of mass of the spot, with the idea of transforming the ROI into an one-dimensional signal with minimum entropy. Then, predictive coding is applied along this path, with a separation between residuals belonging to the spot area and those belonging to the background area. Faramarzpour's method does not allow lossy to lossless coding, although a lossy algorithm has been also proposed. In this case, the spots are adaptively segmented, circle-to-square transformed and joined together, and finally DCT (discrete cosine transform) transformed and quantized. Hence, information loss exists in the circle assignment and in the quantization of the DCT coefficients.

More recently, Lonardi *et al.* [17] proposed lossless and lossy compression algorithms for microarray images (MicroZip). The method uses a fully automatic gridding procedure for separating spots from the background (which can be lossy compressed) and the information is encoded using arithmetic coding. The gridding algorithm is very similar to that of Faramarzpour's method, and is of fundamental importance to the image segmentation step which is performed prior to compression. Through segmentation, the image is split into two channels: foreground and background. Then, for entropy coding, each channel is divided into two 8 bit sub-channels and arithmetic encoded, with the option of being previously processed by a Burrows-Wheeler transform. MicroZip supports a lossy mode in which SPIHT (Set Partitioning In Hierarchical Trees, [18]) is applied to the values of the background least significant sub-channel.

2.2 Standard methods

JBIG, JPEG-LS and JPEG2000 are state-of-the-art standards for coding digital images. They have been developed with different goals in mind, being JBIG more focused on bi-level imagery, JPEG-LS dedicated to the lossless compression of continuous-tone images and JPEG2000 designed with the aim of providing a wide range of functionalities. Next, we give a brief overview of each of these three standards.

JBIG (Joint Bi-level Image Experts Group) was issued in 1993 by ISO/IEC (International Organization for Standardization / International Electrotechnical Commission) and ITU-T (Telecommunication Standardization Sector of the International Telecommunication Union) for the progressive lossless compression of binary and low-precision gray-level images (typically, having less than 6 bits per pixel). The major advantages of JBIG over other existing standards, such as FAX Group 3/4, are its capability of progressive encoding and its superior compression efficiency [5, 6, 19, 20]. The core of JBIG is an adaptive context-based arithmetic coder, relying on 1024 contexts when operating in sequential mode or on low resolution layers of the progressive mode, or 4096 contexts when encoding high resolution layers. More recently, a new version, named JBIG2, has been published [21], introducing additional functionalities to the standard, such as multi-page document compression, two modes of progressive compression, lossy compression and differentiated compression methods for different regions of the image (e.g., text or half-tones) [6].

JPEG-LS was developed by the Joint Photographic Experts Group (JPEG) with the aim of providing a low complexity lossless image standard that could be able to offer better compression efficiency than lossless JPEG

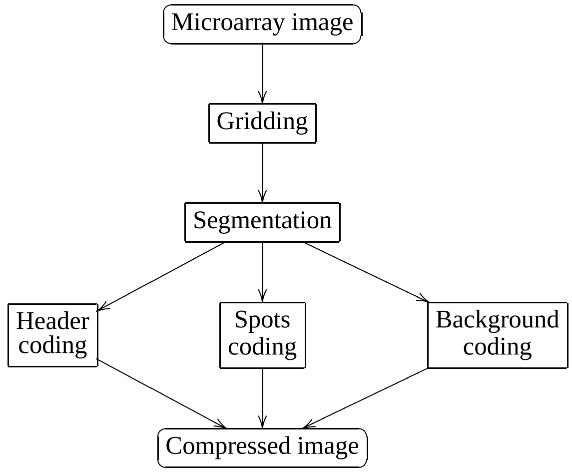

Figure 1. Flow chart of most of the current specialized microarray image compression methods.

[4, 7, 8]. Part 1 of this standard was finalized in 1999. The core of JPEG-LS is based on the LOCO-I algorithm, that relies on prediction, residual modeling and context-based coding of the residuals. Most of the low complexity of this technique comes from the assumption that prediction residuals follow a two-sided geometric probability distribution and from the use of Golomb codes which are known to be optimal for this kind of distributions. Besides lossless compression, JPEG-LS also provides a lossy mode where the maximum absolute error can be controlled by the encoder. This is known as near-lossless compression or L_∞-constrained compression.

From the three image coding standards addressed here, JPEG2000 is the most recent one [2, 4] (Part 1 was published as an International Standard in the year 2000). This standard is based on wavelet technology and EBCOT coding of the wavelet coefficients, providing very good compression performance for a wide range of bit-rates, including lossless coding. Moreover, JPEG2000 allows the generation of embedded code streams, meaning that from a higher bit-rate stream it is possible to extract lower bit-rate instances without the need for re-encoding. This property is of fundamental importance for progressive transmission, for example, over slow communication channels.

These three standard image encoders cover a great variety of coding approaches. In fact, whereas JPEG2000 is transform based, JPEG-LS relies on predictive coding, and JBIG relies on context-based arithmetic coding.

3 Proposed method

In this paper, we present a lossless compression method for microarray images using arithmetic coding with a 3D context model. This method is based on EIDAC [22], which is a compression method used with success for coding simple images, i.e., images with a reduced number of intensity levels. Here, we show that the same idea can be used with microarray images, which in fact are characterized by having a very large number of different intensity levels.

The images are compressed on a bit-plane basis, starting from the most significant bit-plane (MSBP) and stopping at the least significant bit-plane (LSBP), or whenever a bit-plane requires more than one bit per pixel for encoding (the rest of the bit-planes are sent uncoded). The causal context model that drives the arithmetic encoder uses pixels both from the bit-plane currently being encoded (C_{intra}) and from the bit-planes already encoded (C_{inter}).

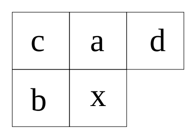

Figure 2. The context configuration used in C_{intra}.

For the C_{intra} part of the context, we use four pixels, as shown in Fig. 2. The number of bits used in C_{inter} depends on the bit-plane and can be as many as 15 bits (see Fig. 3). During the encoding of the image, the number of pixels in the bit-plane above the current and in the MSBP that are used to construct the C_{inter} part of the context decreases, in order to maintain the 19 bit limit of context size (for example, when processing bit-plane number 16, only one pixel of each bit-plane above the actual is used in C_{inter}). This limitation is imposed in order to avoid, on one hand, the increase of memory usage and, on the other hand, the effect of context dilution. Since the decoder is able to select the same context configuration as the encoder, there is no need for side information.

It has been noted by Jörnsten *et al.* that, in general, the eight least significant bit-planes of cDNA microarray images are close to random and, therefore, incompressible [12]. Moreover, this may result in some degradation in the compression performance of some encoders. Since the proposed algorithm uses information of the upper bit-planes, it can encode more efficiently the least significant bit-planes.

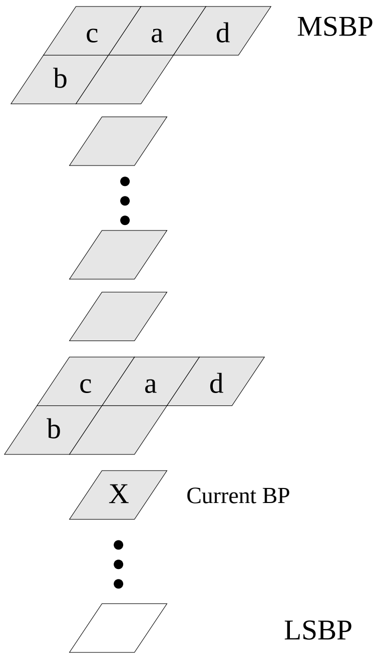

Figure 3. The context configuration used in C_{inter}. In our implementation, in order to limit the maximum number of context bits to 19 ($C_{intra} + C_{inter}$), the number of pixels used in the bit-plane above the current bit-plane and the number of pixels used in the MSBP decrease as encoding progresses from bit-plane to bit-plane.

Our experiments showed that, using our method, this effect is verified only for a small number of bit-planes of some images. Nevertheless, we introduced a simple mechanism that overcomes this problem. As the method proceeds encoding the image, the average bit-rate obtained after encoding each bit-plane is monitored. If, for some bit-plane, the average bit-rate exceeds one bit per pixel, then we stop the encoding process, and the remaining bit-planes are saved without compression.

4 Experimental results

In order to perform the experiments reported in this paper, we collected microarray images from three different publicly available sources: (1) 32 images that we refer to as the Apo AI set and which have been collected from `http://www.stat.berkeley.edu/users/terry/zarray/Html/index.html` (this set was previously used by Jörnsten *et al.* [10–12]); (2) 14 images forming the ISREC set which have been collected from `http://www.isrec.isb-sib.ch/DEA/module8/P5_chip_image/images/`; (3) three images previously used to test MicroZip [17], which were collected from `http://www.cs.ucr.edu/~yuluo/MicroZip/`.

JBIG compression was obtained using version 1.6 of the JBIG Kit package[1], with sequential coding (-q flag). JPEG2000 lossless compression was obtained using version 5.1 of the JJ2000 codec with default parameters (lossless compression)[2]. JPEG-LS coding was obtained using version 2.2 of the SPMG JPEG-LS codec with default parameters[3]. For additional reference, we also give compression results using the popular compression tool GZIP (version 1.2.4).

Table 1 shows the compression results, in number of bits per pixel (bpp), where the first group of images corresponds to the Apo AI set, the second to the ISREC set and the third one to the MicroZip image set. Image size ranges from 1000×1000 to 5496×1956 pixels, i.e., from uncompressed sizes of about 2 megabytes to more than 20 megabytes (all images have 16 bits per pixel). The average results presented take into account the different sizes of the images, i.e., they correspond to the total number of bits divided by the total number of image pixels.

Table 1 shows that, for all the images used in the test, the proposed method is the best among all presented methods. Moreover, the best results obtained with MicroZip [17] in the images ”array1”, ”array2” and ”array3” (MicroZip image set) was, respectively, 11.49, 9.57 and 8.47 bits/pixel. Our method provides compression gains of 2.5%, 10% and 8%, respectively.

5 Conclusion

In this paper, we presented an efficient method for lossless compression of microarray images, allowing progressive, lossy to lossless, decoding. The results obtained show that the proposed method has better compression performance for all images in the test set than the image coding standards used for comparison. Globally, our method is 3.5% better than JPEG-LS, the best of the image coding standards presented in this paper, 5.1% better than JBIG and 7.5% better than lossless JPEG2000. It is important to note that JPEG-LS does not provide progressive decoding, a characteristic that is intrinsic to our method and also provided by JPEG2000 and JBIG. This might be an important functionality if large databases have to be accessed remotely.

The method described in this paper does not require gridding and/or segmentation stages as most of the specialized methods that have been proposed do. This may be an advantage if only compression is sought, since it reduces the computational complexity of the method.

Finally, we should note that we also provide detailed results regarding the lossless compression of microarrays images by state-of-the-art image coding standards, which might facilitate future comparisons by other researchers.

---

[1] `http://www.cl.cam.ac.uk/~mgk25/jbigkit/`.

[2] `http://jj2000.epfl.ch`.

[3] The original website of this codec, `http://spmg.ece.ubc.ca`, is currently unavailable. However, it can be obtained from `ftp://www.ieeta.pt/~ap/codecs/jpeg_ls_v2.2.tar.gz`.

Table 1. Compression results, in bits per pixel (bpp), using lossless JPEG2000, JBIG, JPEG-LS and the proposed method. For reference, results are also given for the popular compression tool GZIP.

Image	Gzip	JPEG2000	JBIG	JPEG-LS	Proposed
1230c1G	13.263	11.864	11.544	11.408	10.961
1230c1R	13.181	11.488	11.226	11.002	10.701
1230c2G	13.198	11.805	11.630	11.463	11.071
1230c2R	13.097	11.424	11.343	11.052	10.856
1230c3G	12.729	11.190	10.879	10.715	10.354
1230c3R	12.483	10.618	10.461	10.143	9.983
1230c4G	12.849	11.272	11.122	10.876	10.518
1230c4R	12.803	10.936	10.854	10.528	10.380
1230c5G	12.531	10.958	10.633	10.452	10.061
1230c5R	12.371	10.488	10.307	9.975	9.797
1230c6G	12.691	11.268	10.962	10.792	10.372
1230c6R	12.721	11.102	10.982	10.696	10.520
1230c7G	12.777	11.130	10.818	10.652	10.272
1230c7R	12.449	10.451	10.316	9.982	9.833
1230c8G	12.874	11.332	11.094	10.884	10.495
1230c8R	12.966	11.204	11.076	10.785	10.596
1230ko1G	12.410	10.766	10.369	10.206	10.020
1230ko1R	12.695	10.979	10.606	10.422	10.303
1230ko2G	12.465	10.852	10.618	10.410	10.097
1230ko2R	12.528	10.768	10.631	10.324	10.169
1230ko3G	12.822	11.309	11.013	10.833	10.476
1230ko3R	12.674	10.925	10.761	10.477	10.261
1230ko4G	12.510	10.976	10.697	10.516	10.136
1230ko4R	12.609	10.887	10.730	10.409	10.251
1230ko5G	12.795	11.286	11.100	10.881	10.477
1230ko5R	12.589	10.874	10.704	10.409	10.197
1230ko6G	12.594	11.086	10.917	10.679	10.289
1230ko6R	12.459	10.659	10.546	10.208	10.067
1230ko7G	12.752	11.278	10.929	10.785	10.371
1230ko7R	12.554	10.772	10.613	10.295	10.083
1230ko8G	12.669	11.173	10.965	10.737	10.338
1230ko8R	12.644	10.889	10.785	10.448	10.285
Average	**12.711**	**11.063**	**10.851**	**10.608**	**10.331**
Def661Cy3	12.658	11.914	11.218	11.713	10.550
Def661Cy5	11.418	9.714	9.451	9.392	8.909
Def662Cy3	11.636	10.881	10.007	10.575	9.221
Def662Cy5	12.722	11.369	11.251	11.156	10.643
Def663Cy3	12.437	11.903	11.023	11.665	10.210
Def663Cy5	11.961	10.405	10.124	10.151	9.620
Def664Cy3	12.322	11.592	10.813	11.384	10.168
Def664Cy5	13.142	11.768	11.755	11.632	11.229
Def665Cy3	13.363	12.462	12.111	12.289	11.511
Def665Cy5	14.451	13.590	13.429	13.557	12.809
Def666Cy3	11.768	10.946	10.132	10.659	9.477
Def666Cy5	13.116	11.727	11.748	11.572	11.140
Def667Cy3	11.690	10.540	9.923	10.248	9.274
Def667Cy5	11.807	10.304	9.951	10.033	9.426
Average	**12.464**	**11.366**	**10.925**	**11.145**	**10.300**
array1	13.385	12.027	11.819	11.590	11.184
array2	11.470	9.272	9.071	8.737	8.599
array3	10.375	8.599	8.351	7.996	7.848
Average	**11.434**	**9.515**	**9.297**	**8.974**	**8.788**
Total Average	**12.273**	**10.653**	**10.393**	**10.218**	**9.856**

References

[1] S. K. Moore, “Making chips to probe genes,” *IEEE Spectrum*, vol. 38, no. 3, pp. 54–60, Mar. 2001.

[2] ISO/IEC, *Information technology - JPEG 2000 image coding system*, ISO/IEC International Standard 15444–1, ITU-T Recommendation T.800, 2000.

[3] A. Skodras, C. Christopoulos, and T. Ebrahimi, “The JPEG 2000 still image compression standard,” *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 36–58, Sept. 2001.

[4] D. S. Taubman and M. W. Marcellin, *JPEG 2000: image compression fundamentals, standards and practice*, Kluwer Academic Publishers, 2002.

[5] ISO/IEC, *Information technology - Coded representation of picture and audio information - progressive bi-level image compression*, International Standard ISO/IEC 11544 and ITU-T Recommendation T.82, Mar. 1993.

[6] D. Salomon, *Data compression - The complete reference*, Springer, 2nd edition, 2000.

[7] ISO/IEC, *Information technology - Lossless and near-lossless compression of continuous-tone still images*, ISO/IEC 14495–1 and ITU Recommendation T.87, 1999.

[8] M. J. Weinberger, G. Seroussi, and G. Sapiro, “The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS,” *IEEE Trans. on Image Processing*, vol. 9, no. 8, pp. 1309–1324, Aug. 2000.

[9] R. Jörnsten, B. Yu, W. Wang, and K. Ramchandran, “Compression of cDNA and inkjet microarray images,” in *Proc. of the 9th IEEE Int. Conf. on Image Processing, ICIP-2002*, Rochester, NY, Sept. 2002, vol. 3, pp. 961–964.

[10] R. Jörnsten and B. Yu, “Compression of cDNA microarray images,” in *Proc. of the IEEE Int. Symp. on Biomedical Imaging, ISBI-2002*, Washington, DC, July 2002, pp. 38–41.

[11] R. Jörnsten, Y. Vardi, and C.-H. Zhang, “On the bit-plane compression of microarray images,” in *Proc. of the 4th Int. L1-norm Conf.*, Y. Dodge, Ed., 2002.

[12] R. Jörnsten, W. Wang, B. Yu, and K. Ramchandran, “Microarray image compression: SLOCO and the effect of information loss,” *Signal Processing*, vol. 83, pp. 859–869, 2003.

[13] J. Hua, Z. Xiong, Q. Wu, and K. Castleman, “Fast segmentation and lossy-to-lossless compression of DNA microarray images,” in *Proc. of the Workshop on Genomic Signal Processing and Statistics, GENSIPS*, Raleigh, NC, Oct. 2002.

[14] J. Hua, Z. Liu, Z. Xiong, Q. Wu, and K. Castleman, “Microarray BASICA: background adjustment, segmentation, image compression and analysis of microarray images,” in *Proc. of the 10th IEEE Int. Conf. on Image Processing, ICIP-2003*, Barcelona, Spain, Sept. 2003, vol. 1, pp. 585–588.

[15] N. Faramarzpour, S. Shirani, and J. Bondy, “Lossless DNA microarray image compression,” in *Proc. of the 37th Asilomar Conf. on Signals, Systems, and Computers, 2003*, Nov. 2003, vol. 2, pp. 1501–1504.

[16] N. Faramarzpour and S. Shirani, “Lossless and lossy compression of DNA microarray images,” in *Proc. of the Data Compression Conf., DCC-2004*, Snowbird, Utah, Mar. 2004, p. 538.

[17] S. Lonardi and Y. Luo, “Gridding and compression of microarray images,” in *Proc. of the IEEE Computational Systems Bioinformatics Conf., CSB-2004*, Stanford, CA, Aug. 2004.

[18] A. Said and W. A. Pearlman, “A new, fast, and efficient image codec based on set partitioning in hierarchical trees,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243–250, June 1996.

[19] H. Hampel, R. B. Arps, C. Chamzas, D. Dellert, D. L. Duttweiler, T. Endoh, W. Equitz, F. Ono, R. Pasco, I. Sebestyen, C. J. Starkey, S. J. Urban, Y. Yamazaki, and T. Yoshida, “Technical features of the JBIG standard for progressive bi-level image compression,” *Signal Processing: Image Communication*, vol. 4, no. 2, pp. 103–111, Apr. 1992.

[20] A. N. Netravali and B. G. Haskell, *Digital pictures: representation, compression and standards*, Plenum, New York, 2nd edition, 1995.

[21] ISO/IEC, *JBIG2 bi-level image compression standard*, International Standard ISO/IEC 14492 and ITU-T Recommendation T.88, 2000.

[22] Y. Yoo, Y. G. Kwon, and A. Ortega, “Embedded image-domain compression using context models,” in *Proc. of the 6th IEEE Int. Conf. on Image Processing, ICIP-99*, Kobe, Japan, Oct. 1999, vol. I, pp. 477–481.