LOSSLESS COMPRESSION OF MICROARRAY IMAGES

António J. R. Neves, Armando J. Pinho

Signal Processing Lab, DET / IEETA
University of Aveiro, 3810-193 Aveiro, Portugal
an@ieeta.pt — ap@det.ua.pt

ABSTRACT

Microarray experiments are characterized by a massive amount of data in the form of images. Since the interest in microarray technology is growing nowadays, a large number of microarray images is currently being produced. In this paper, we present a lossless method for efficiently compress microarray images based on arithmetic coding using a 3D context model. Our method produces an embedded bit-stream that allows progressive decoding. We present the compression results using a large set of images and compare these results with three image coding standards, namely, lossless JPEG2000, JBIG and JPEG-LS. We also compare our results with the two most recent specialized methods, using a common set of images. The proposed method gives better results for all images of the test set.

***Index Terms*—** Image coding, biomedical image processing, data compression.

1. INTRODUCTION

In a microarray experiment, a matrix of spots is arranged on a slide of glass or nylon. The result of this experiments 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 [1]. These images are analyzed, resulting in the calculation of the gene expression level for each spot. 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.

Giving the massive amount of data currently produced, and the need of long-term storage and efficient transmission, the development of an efficient compression method 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. Moreover 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.

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

In this paper, we present a compression method based on arithmetic coding driven by a 3D context model (preliminary results have been presented in [2]). Basically, 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 [3, 4], JBIG [5, 6] and JPEG-LS [7, 8], and also with the two most recent specialized methods, MicroZip [9] and Zhang's method [10]. A detailed study on the use of standards for compression of microarray images can be found in [11].

2. CODING METHODS FOR MICROARRAY IMAGES

To the best of our knowledge, at the time of writing, there are five published methods for the lossy and / or lossless compression of microarray images, namely, the works of Jörnsten *et al.* [12, 13, 14, 15], Hua *et al.* [16, 17], Faramarzpour *et al.* [18, 19], Lonardi *et al.* [9] and Zhang *et al.*[10]. Although different, they share, nevertheless, a common structure. All the methods perform griding followed by segmentation and then encode the background and the spots separately. The segmentation information is also included.

It is often difficult to compare the performance among the methods 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 provide compression results on a set of 49 images gathered from three different publicly available sources and compare the results obtained with three standard image coding techniques, namely, lossless JPEG2000[1] [3, 4], JBIG[2] [5, 6] and JPEG-LS[3] [7, 8].

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.

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. For a reference on the use of standards for compression of microarray images see [11].

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 was inspired on EIDAC [20], which is a compression method used with success for coding simple images.

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}), see Fig. 1, and from the bit-planes already encoded (C_{inter}), see Fig. 2.

When encoding the eight least significant bit-planes, the context model is only formed with pixels from the upper bit-planes, Fig. 2 (b). This procedure is done to avoid the degradation in compression due to, in general, the eight least significant bit-planes being close to random and, therefore, almost incompressible [15]. We have also introduced another mechanism to 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.

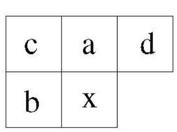

Fig. 1. The context configuration used in C_{intra}. This context is not used for encoding the eight least significant bit-planes.

[1] http://jj2000.epfl.ch.

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

[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.

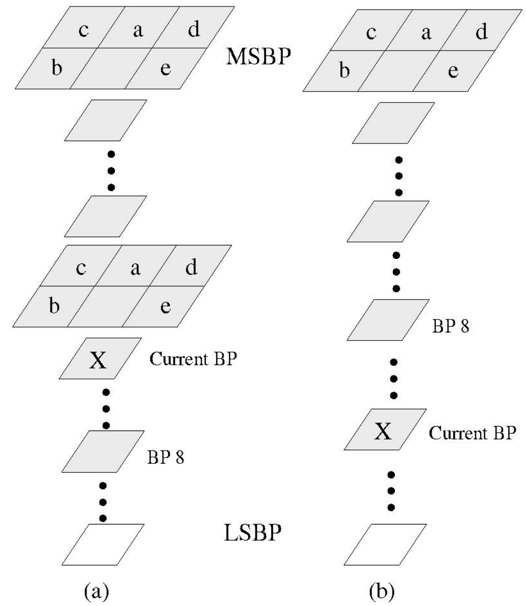

Fig. 2. Context configuration used in C_{inter}. (a) Context configuration used when the 8 most significant bit-planes are being encoded. (b) When encoding the eight least significant bit-planes the information of the bit-plane immediately above the current is discarded.

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.* [14, 13, 15]); (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[9] and Zhang's method[10], which were collected from http://www.cs.ucr.edu/~yuluo/MicroZip/.

Table 2. Compression results, in bits per pixel (bpp), using MicroZip, Zhang's method and the proposed method.

Image	MicroZip	Zhang	Proposed
array1	11.490	11.380	11.125
array2	9.570	9.260	8.640
array3	8.470	8.120	7.980
Average	**9.532**	**9.243**	**8.840**

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 and the second to the ISREC set. Table 2 shows the compression results for 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 tested methods. Table 2 confirms the performance of our method relatively to the two newest specialized methods for compression of microarray images. Our method provides compression gains of 7.3% relatively to MicroZip and 4.4% relatively to Zhang's method.

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 3.8% better than JPEG-LS, the best of the image coding standards presented in this paper, 5.5% better than JBIG and 7.7% 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 to JPEG2000 and JBIG. This might be an important functionality if large databases have to be accessed remotely. Comparing to the specialized methods, our method is the best for all images in the MicroZip set.

Finally, the method described in this paper does not require griding 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.

6. REFERENCES

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

[2] A. J. R. Neves, A. J. Pinho, and A. R. C. Paiva", "Lossless bit-plane compression of microarray images using 3D context models", in *Proc. of the 5th IASTED Int. Conf. on Visualization, Imaging, and Image Processing, VIIP-2005*, Benidorm, Spain, Sept. 2005, pp. 253–258.

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

[4] 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.

[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] 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.

[10] Yong Zhang, Rahul Parthe, and Don Adjeroh, "Lossless compression of DNA microarray images," in *Proc. of the IEEE Computational Systems Bioinformatics Conference, CSB-2005*, Stanford, CA, Aug. 2005.

[11] A. J. Pinho, A. R. C. Paiva, and A. J. R. Neves, "On the use of standards for microarray lossless image compression" *IEEE IEEE Trans. on Biomedical Engineering"*, vol. 53, no. 3, pp. 563–566, Mar. 2006.

[12] 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.

[13] 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.

[14] R. Jörnsten, Y. Vardi, and C.-H. Zhang, "On the bitplane compression of microarray images," in *Proc. of the 4th Int. L1-norm Conf.*, Y. Dodge, Ed., 2002.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

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	10.896
1230c1R	13.181	11.488	11.226	11.002	10.642
1230c2G	13.198	11.805	11.630	11.463	11.011
1230c2R	13.097	11.424	11.343	11.052	10.800
1230c3G	12.729	11.190	10.879	10.715	10.288
1230c3R	12.483	10.618	10.461	10.143	9.940
1230c4G	12.849	11.272	11.122	10.876	10.469
1230c4R	12.803	10.936	10.854	10.528	10.341
1230c5G	12.531	10.958	10.633	10.452	10.002
1230c5R	12.371	10.488	10.307	9.975	9.756
1230c6G	12.691	11.268	10.962	10.792	10.309
1230c6R	12.721	11.102	10.982	10.696	10.474
1230c7G	12.777	11.130	10.818	10.652	10.203
1230c7R	12.449	10.451	10.316	9.982	9.793
1230c8G	12.874	11.332	11.094	10.884	10.439
1230c8R	12.966	11.204	11.076	10.785	10.540
1230ko1G	12.410	10.766	10.369	10.206	9.978
1230ko1R	12.695	10.979	10.606	10.422	10.272
1230ko2G	12.465	10.852	10.618	10.410	10.044
1230ko2R	12.528	10.768	10.631	10.324	10.135
1230ko3G	12.822	11.309	11.013	10.833	10.409
1230ko3R	12.674	10.925	10.761	10.477	10.196
1230ko4G	12.510	10.976	10.697	10.516	10.077
1230ko4R	12.609	10.887	10.730	10.409	10.215
1230ko5G	12.795	11.286	11.100	10.881	10.427
1230ko5R	12.589	10.874	10.704	10.409	10.149
1230ko6G	12.594	11.086	10.917	10.679	10.232
1230ko6R	12.459	10.659	10.546	10.208	10.040
1230ko7G	12.752	11.278	10.929	10.785	10.298
1230ko7R	12.554	10.772	10.613	10.295	10.036
1230ko8G	12.669	11.173	10.965	10.737	10.275
1230ko8R	12.644	10.889	10.785	10.448	10.247
Average	**12.711**	**11.063**	**10.851**	**10.608**	**10.280**
Def661Cy3	12.658	11.914	11.218	11.713	10.406
Def661Cy5	11.418	9.714	9.451	9.392	8.874
Def662Cy3	11.636	10.881	10.007	10.575	9.161
Def662Cy5	12.722	11.369	11.251	11.156	10.555
Def663Cy3	12.437	11.903	11.023	11.665	10.121
Def663Cy5	11.961	10.405	10.124	10.151	9.528
Def664Cy3	12.322	11.592	10.813	11.384	10.001
Def664Cy5	13.142	11.768	11.755	11.632	11.138
Def665Cy3	13.363	12.462	12.111	12.289	11.436
Def665Cy5	14.451	13.590	13.429	13.557	12.663
Def666Cy3	11.768	10.946	10.132	10.659	9.305
Def666Cy5	13.116	11.727	11.748	11.572	11.043
Def667Cy3	11.690	10.540	9.923	10.248	9.218
Def667Cy5	11.807	10.304	9.951	10.033	9.331
Average	**12.464**	**11.366**	**10.925**	**11.145**	**10.199**
Total Average	**12.640**	**11.141**	**10.872**	**10.762**	**10.257**