LOSSLESS MICROARRAY IMAGE COMPRESSION USING REGION BASED PREDICTORS

A. Neekabadi[1]*, S. Samavi*[1]*, S.A. Razavi*[1]*, N. Karimi*[1]*, S. Shirani*[2]

Department of Electrical and Computer Engineering
[1]Isfahan University of Technology, Isfahan, Iran
[2]McMaster University, Hamilton, Canada

ABSTRACT

Microarray image technology is a powerful tool for monitoring the expression of thousands of genes simultaneously. Each microarray experiment produces large amount of image data, hence efficient compression routines that exploit microarray image structures are required. In this paper we introduce a lossless image compression method which segments the pixels of the image into three categories of background, foreground, and spot edges. The segmentation is performed by finding a threshold value which minimizes the weighted sum of the standard deviations of the foreground and background pixels. Each segment of the image is compressed using a separate predictor. The results of the implementation of the method show its superiority compared to the well-known microarray compression schemes as well as to the general lossless image compression standards.

Index Terms— *microarray, lossless image compression.*

1. INTRODUCTION

Microarrays have become an important tool for understanding of gene function, regulation and interaction through the simultaneous study of thousands of genes. The output of a single microarray experiment is a pair of 16 bits per pixels (bpp) digital images whose total size is typically in the order of tens of MB. 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 [1]. Figure 1 shows a part of a typical microarray image.

Both lossy and lossless methods have been employed for compression of microarray images. It is more desirable to keep the images in a lossless format due to the fact that the existing analytical methods for these images are in their developing stages. In other words, it seems wise to keep the microarray images free of losses, in order to facilitate future re-analysis by better algorithms.

JPEG-LS [2], JPEG2000 [3] and JBIG [3] are state-of-the art methods for coding digital images. They have been developed for different purposes, that is 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.

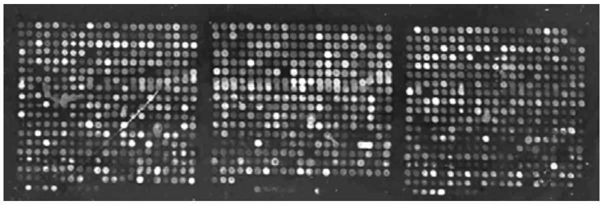

Figure.1. Part of a typical DNA microarray image.

Different methods have been proposed specifically for compressing microarray images. In [4] an algorithm called SLOCO was proposed. This is a simple extension of LOCO-I [3], the basic algorithm used in JPEG-LS. SLOCO also considers denoising, lossy compression, and methods of fast access to individual spots. In reference [5], a spiral scanning method is proposed based on the circular nature of the spots where pixels are predicted by their immediate neighbor in the scanning path. Hua et.al. [6], proposed wavelet-based lossy and lossless compression schemes for microarray images. Lonardi [1] proposes an algorithm called MicroZip where arithmetic coding (AC) and the Burrows–Wheeler transform (BWT) are used for lossless compression of microarray data. This is done after dividing the pixel values into their least significant bits (LSB) and most significant bits (MSB). In [7] a context-based method is proposed for lossless compression of microarray images using prediction by partial approximate matching (PPAM). Their method produced superior results in lossless compression as compared to any other method. Samavi et.al. [8] performed microarray image compression by pseudo RLE coding through real-time hardware architecture hence they did not achieve high compression ratios.

Among the recent lossless microarray compression methods only MicroZip and PPAM have been able to get better results than JPEG-LS when applied to standard microarray images [7]. In this paper, we present a new lossless method for efficient compression of microarray images which produces results that are better than Microzip and PPAM. Our method is based on categorization of image pixels in three groups of foreground, background, and spot boundary. A thresholding scheme is applied which minimizes the weighted sum of the standard deviations of the foreground and background regions. In the proposed scheme each segmented region is separately compressed with different predictors.

This paper is organized in the following manner. In section 2 the details of the proposed method are presented. Simulation results are discussed in section 3 and concluding remarks are offered in section 4 of the paper.

2. PROPOSED METHOD

Most successful lossless image compression algorithms are context-based and they exploit the two-dimensional spatial redundancy in natural images [3, 8]. For microarray images using compression schemes based on predictions in the spatial domain have difficulties with the many high intensity spots. Such problems can be overcome by encoding the spots and the background separately. At first, a segmentation map (mask) separating the spots from the background is generated, and then the image is compressed.

In this paper a compression scheme based on prediction in the spatial domain is offered. In this method besides separating the foreground from background a third region of spots boundaries is defined. Predictor based methods produce large errors at the spot's boundary. We are able to circumvent this problem by introducing the third region. Each of these three regions are separately predicted and coded. In other methods the prediction of spots' edges produces large errors. In our method the edges are detected and hence separate prediction for them is possible. This extra prediction causes no extra overhead for the compression algorithm.

The block diagram of Figure 2 shows the overall structure of the proposed algorithm. First step is to perform segmentation. Foreground is separated from the background through the application of a threshold that is found from the standard deviations of the intensities of these two regions. The segmentation unit also divides the foreground region into edges and spot regions. Hence, the output of the segmentation unit is the three mentioned masks. Each of the three compression units of Figure 2 performs two-dimensional prediction as well as performing a statistical coding routine. There is also a mask compressor dedicated to compression of the segmentation map. In the following subsections the details of each unit is explained.

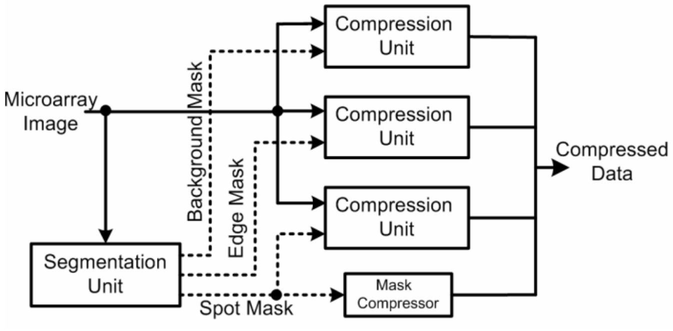

Figure 2. Main functional units of the proposed algorithm.

2.1. Segmentation

The first step in our proposed compression algorithm is the segmentation of the image into three distinct regions. We propose a dynamic thresholding scheme for the segmentation purpose. By applying a threshold value the pixels of the image are grouped into two sets. The number of pixels in each set and the standard deviation of the intensities of these pixels can be found. This process can be applied to all possible threshold values. We need to select one of these threshold values. Equation (1) gives us the desired threshold. This threshold guarantees that the weighted sum of the standard deviation of the background and foreground is minimal.

$$Threshold = \underset{0 \le T \le 2^{16}-1}{\operatorname{argmin}} \{f(T)\}$$
$$f(T) = \mathrm{Std}(B_T) \times \mathrm{size}(B_T) + \mathrm{Std}(F_T) \times \mathrm{size}(F_T) \qquad (1)$$
$$B_T = \{p \in Image \mid p < T\}\ ,\ F_T = \{p \in Image \mid p \ge T\}$$

In Equation (1) B_T and F_T are respectively the sets of pixels in the background and foreground of the image after the application of a threshold, T. All pixels with intensities above T are grouped as foreground pixels. *Std* and *size* functions respectively find the standard deviation and the number of all pixels in a region.

The plot of Figure 3 shows the *f(T)* function for a specific microarray image. Relatively similar plots are obtained for other microarray images, too. It is apparent that the function plunges down at a certain threshold value which is chosen as the final threshold value. It is worth mentioning that instead of testing all possible threshold values to find the minimum value of *f(T)* we used a recursive search algorithm which accelerates the search routine.

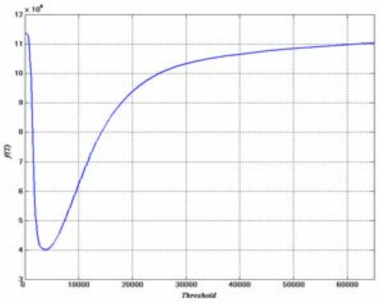

Figure 3. Plot of *f(T)* for a range of possible threshold values.

A binary map is built after the thresholding processes. Foreground pixels are represented by 1's in the map and 0's mean background pixels. In order to eliminate isolated points in the mask an erosion operation is performed on the binary map. This process is shown in Figure 4. In order to outline the spot's edge a morphological dilation is performed on the output of the erosion step and the result is subtracted from the outcome of the erosion step. The result of this subtraction gives the edge mask. A 3×3 square structuring element is used for the morphological operations. Therefore three separate binary masks are produced. With the masks at hand, the whole image is segmented into three regions of the foreground, the background, and the spots boundaries. These three masks are used in the compression units of Figure 2.

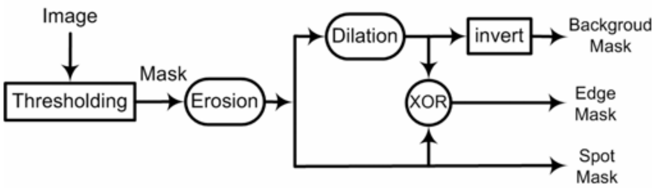

Figure 4. Block diagram of the segmentation unit.

Figure 5 shows the results for the mask generation and the segmentation performed on a part of a microarray image. Part (a) of Figure 5 shows the original image. Parts b, c, and d of Figure 5 show the background mask, the spot mask, and the edge mask. As can be seen in Figure 4, edge and background masks can be reconstructed from the spot mask. Therefore, only the spot mask needs to be compressed and saved. To compress the spot mask first RLE and then Huffman coding is performed on it. An average of 0.1 bpp is achieved for this type of mask compression.

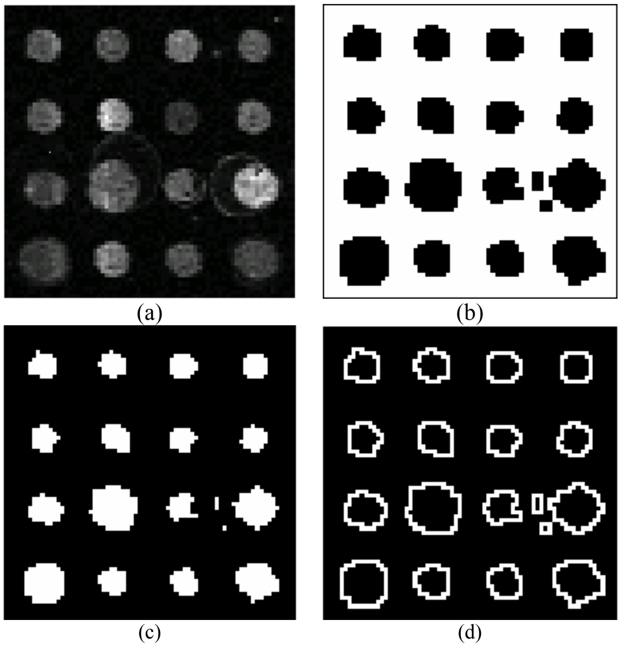

Figure 5. (a) Original image, (b) Background mask, (c) Spot mask, (d) Edge mask.

2.2. Compression

In this section we explain the compression blocks that are shown in Figure 2. The pixels from each segment are independently compressed by using different predictors. Figure 6 shows the block diagram of the compression scheme. The image is scanned from top-left towards bottom-right in a row by row manner. The predictor of Figure 7 is placed on each pixel of the image.

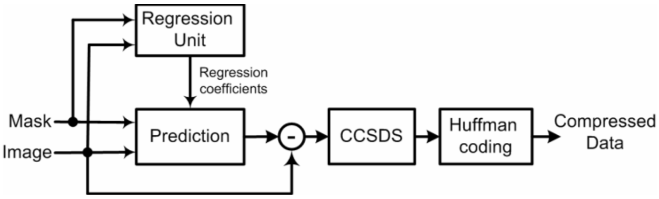

Figure 6. Block diagram of the compression unit.

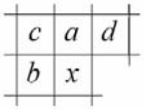

Figure 7. Pixels used in prediction of x.

The function of this predictor depends on segment that the pixel belongs to. This function is also dependent on the location of the pixel inside the segment. Equation 2 describes the function of the predictor.

$$\hat{x} = \lfloor k \times m \rfloor \tag{2}$$

In Equation 2, $\hat{x}$ is the predicted value for pixel x. Also, m is the mean of the neighbors of x, according to Figure 7, that are present in the same segment as x. The averaging is only performed on those neighbors that belong to the same segment as x. In Equation 2 variable k is the robust linear regression coefficient [9]. The use of this coefficient guarantees that the sum of squared errors is minimized. The image is once scanned to find the best value for k for each predictor. Then in the second scan these k's are used to perform the prediction. The three values of k are need for reconstruction of the image. The CCSDS (Consultative Committee for Space Data Systems) [10] algorithm is applied to the produced prediction errors. This algorithm ensures that these signed values are all turned into positive numbers. Finally, Huffman coding is applied to the output of the CCSDS algorithm.

3. EXPERIMENTAL RESULTS

In order to show the functionality and accuracy of our method we used microarray images from three publicly available resources. The first group containing a set of 32 microarray images was obtained from [www.stat.berkeley.edu/users/terry/zarray/Html/],the second set was from [www.isrec.isbsib.ch/DEA/module8/P5_chip _image/Images/] and the third group was MicroZip images that are accessible at [www.cs.ucr.edu/yuluo/MicroZip/]. Table 1 shows the compression results in number of bits per pixel (bpp), for three standard compression tools and the proposed method. The results of the standard methods of lossless JPEG2000, JBIG, and JPEG-LS are obtained from [11]. These three standard image encoders cover a great variety of coding approaches. This diversity in coding engines is helpful when drawing conclusions regarding the usability of each of these methods for the microarray image compression. We did not list all of the results instead averages are shown for each group of images. 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 algorithm produces the smallest bpp among all presented methods. Overall, our method is 3.9% better than JPEG-LS, the leading lossless image coding standard, 5.5% better than JBIG and 7.8% better than lossless JPEG2000. The performance of the proposed algorithm depends on the presence of noise in the image. For example for the ISREC images (the second set of images), which have low noise presence, the proposed method performed 8.4% better than JPEG-LS, 6.6% better than JBIG, and 10.2% better than JPEG2000.

Table 1. Comparison with standard methods (bpp).

Image		JPEG 2K	JBIG	JPEG-LS	Proposed
1	1230c1G	11.864	11.544	11.408	10.761
2	1230c1R	11.488	11.226	11.002	10.507
3	1230c2G	11.805	11.630	11.463	11.071
4	1230c2R	11.424	11.343	11.052	10.856
:	:	:	:	:	:
31	1230ko8G	11.173	10.965	10.737	10.322
32	1230ko8R	10.889	10.785	10.448	10.140
Average		**11.063**	**10.851**	**10.608**	**10.250**
1	Def661Cy3	11.914	11.218	11.713	10.39
2	Def661Cy5	9.714	9.451	9.392	8.929
3	Def662Cy3	10.881	10.007	10.575	9.221
4	Def662Cy5	11.369	11.251	11.156	10.643
:	:	:	:	:	:
13	Def667Cy3	10.540	9.923	10.248	9.180
14	Def667Cy5	10.304	9.951	10.033	9.304
Average		**11.366**	**10.925**	**11.145**	**10.202**
1	array1	12.027	11.819	11.590	11.006
2	array2	9.272	9.071	8.737	8.725
3	array3	8.599	8.351	7.996	7.957
Average		**9.515**	**9.297**	**8.974**	**8.856**
Total Average		**10.653**	**10.393**	**10.218**	**9.816**

Shown in Table 2 are the results for a number of methods that are specifically designed for the lossless microarray image compression. Also shown in Table 2 are the results of our method for the third group of images. Performance of our method was 7.7% better than that of MicroZip(AC) [1], 7% better than MicroZip(BWT) [1], and 4.2% better than PPAM [7]. It needs mentioning that the many of microarray compression methods such as SLOCO[4] and BASICA[6] produce inferior results as compared to JPEG-LS, hence they are not listed in our comparison tables. Also, the compression method used in [12] is specifically designed for microarray images and produces lower compression ratios than the proposed method; hence we did not include their results in the comparison table.

Table 2. Comparison of various microarray algorithms (bpp).

Image	MicroZip (AC) [1]	MicroZip (BWT) [1]	PPAM [7]	Proposed
array1	11.69	11.49	11.38	11.006
array2	9.75	9.57	9.26	8.725
array3	8.32	8.47	8.12	7.957
Average	**9.60**	**9.53**	**9.24**	**8.856**

4. CONCLUSION

In this paper, we presented an efficient method for lossless compression of microarray images. This method is based on the idea of segmenting the image into three distinct fields of spots, background, and spots' edges. We were able to reduce the magnitude of the prediction errors that are present at the spots' edges. The proposed method has better compression performance (for all images in the test set) than the image coding standards used for comparison. PPAM and MicroZip that are the best-known microarray compression methods produced inferior results as compared to the proposed method.

5. REFERENCES

[1] Lonardi and Y. Luo, "Gridding and compression of microarray images", *IEEE Computer Society Bioinformatics Conf.*, Aug. 2004.

[2] 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.

[3] D. Solomon, *Data compression: the complete reference*, 3'Th Edition, Springer, 2004.

[4] R. Jornsten, W. Wang, B. Yu, and K. Ramchandran, "Microarray image compression: SLOCO and the effect of information loss", *Signal Processing*, vol. 83, Elsevier , 2003.

[5] N. Faramarzpour, S. Shirani, and J. Bondy, "Lossless DNA microarray image compression", *Proc. of the 37th Asilomar Conf. on Signals, Systems, and Computers*, Nov. 2003.

[6] J. Hua, Z. Liu, Z. Xiong, Q. Wu, K. Castleman, "Microarray BASICA: Background adjustment, segmentation, image compression and analysis of microarray images", *Proceedings of the International Conference on Image Processing*, vol. 1, 2003.

[7] D.A. Adjeroh, Y. Zhang, R. Parthe, "On denoising and compression of DNA microarray images," *Pattern Recognition* ,Volume 39, Issue 12, Pages 2478-2493, Elsevier, December 2006.

[8] S. Samavi, S. Shirani, and N. Karimi, "Real-time processing and Compression of DNA microarray images", *IEEE Trans. on image processing*, Vol.15, pp754-766, 2006.

[9] P.W Holland, and R.E. Welsch, "Robust Regression Using Iteratively Reweighted Least-Squares," *Communications in Statistics: Theory and Methods*, A6, 1977.

[10] K. Sayood, *Introduction to Data Compression*, Second Edition, Morgan Kaufmann, 2000.

[11] A.J. Pinho, A.R.C. Paiva and A.J.R. Neves, "On the use of standards for microarray lossless image compression", *IEEE Trans. on Biomedical Engineering*, vol. 53, March 2006.

[12] A.J.R Neves and A.J. Pinho, "Lossless Compression of Microarray Images", *IEEE International Conference on Image Processing (ICIP)*, pp 2505-2508, Oct. 2006.