[7] *3SPACE FASTRAK User's Manual*, Polhemus Incorporated, Colchester, VTt, 2002.
[8] R. M. Leahy, J. C. Mosher, M. E. Spencer, M. X. Huang, and J. D. Lewine, "A study of dipole localization accuracy for MEG and EEG using a human skull phantom," *Electroencephalogr. Clin. Neurophysiol.*, vol. 107, no. 2, pp. 159–173, Aug 1998.
[9] D. L. Collins, A. P. Zijdenbos, V. Kollokian, J. G. Sled, N. J. Kabani, C. J. Holmes, and A. C. Evans, "Design and construction of a realistic digital brain phantom," *IEEE Trans. Med. Imag.*, vol. 17, no. 3, pp. 463–468, Jun. 1998.
[10] J. Malmivuo and R. Plonsey, *Bioelectromagnetism—Principles and Applications of Bioelectric and Biomagnetic Fields*. New York: Oxford Univ. Press, 1995.

# On the Use of Standards for Microarray Lossless Image Compression

Armando J. Pinho*, António R. C. Paiva, and António J. R. Neves

***Abstract*—The interest in methods that are able to efficiently compress microarray images is relatively new. This is not surprising, since the appearance and fast growth of the technology responsible for producing these images is also quite recent. In this paper, we present a set of compression results obtained with 49 publicly available images, using three image coding standards: lossless JPEG2000, JBIG, and JPEG-LS. We concluded that the compression technology behind JBIG seems to be the one that offers the best combination of compression efficiency and flexibility for microarray image compression.**

***Index Terms*—Image coding standards, JBIG, JPEG-LS, JPEG2000, lossless image compression, lossy-to-lossless compression, microarray images.**

## I. Introduction

The raw data resulting from a DNA microarray experiment [1] is typically conveyed by two images of 16 bits per pixel (bpp), 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 [2].

In this paper, we report a set of experiments that have been performed with the aim of providing a reference regarding the performance of standard image coding techniques, namely, lossless JPEG2000 [3], [4] JBIG [5], and JPEG-LS [4], [6], when applied to the lossless compression of microarray images. In fact, although a number of new techniques has already been proposed for microarray image compression, as we show in Section II, it is often difficult to compare the performance among them and/or in relation to current standards. One of the factors that contributes to this limitation is the lack of results regarding a common and representative set of images. Moreover, and in order to facilitate future comparisons by other researchers, these images should be publicly available. In this paper, we tried to overcome some of these drawbacks, by providing compression results on a set of 49 images gathered from three different publicly available sources.

Another objective of this work was that of trying to identify compression technologies that, on one hand, provide efficient lossless compression results and, on the other hand, offer relevant features for the microarray image compression problem, such as lossy-to-lossless reconstruction. From the three image coding standards that we addressed, we have been able to identify JBIG as potentially the most interesting.

Manuscript received December 16, 2004; revised June 28, 2005. This work was supported in part by the Fundação para a Ciência e a Tecnologia (FCT) and by the INFOBIOMED European NoE. *Asterisk indicates corresponding author.*

*A. J. Pinho is with the Department de Electrónica e Telecomunicações and with the Instituto de Engenharia Electrónica e Telemática de Aveiro (IEETA), Universidade de Aveiro, 3810-193 Aveiro, Portugal (e-mail: ap@det.ua.pt).
A. R. C. Paiva was with the Instituto de Engenharia Electrónica e Telemática de Aveiro (IEETA), Universidade de Aveiro, 3810-193 Aveiro, Portugal, and is now with the University of Florida, Gainesville, FL 32611 USA (e-mail: arpaiva@ufl.edu).
A. J. R. Neves is with the Department de Electrónica e Telecomunicações and with the Instituto de Engenharia Electrónica e Telemática de Aveiro (IEETA), Universidade de Aveiro, 3810-193 Aveiro, Portugal (e-mail: an@ieeta.pt).

Digital Object Identifier 10.1109/TBME.2005.869782

## II. 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.* [2], Hua *et al.* [7], Faramarzpour *et al.* [8], and Lonardi *et al.* [9]. Next, we give a brief overview of each of these techniques.

The technique proposed by Jörnsten *et al.* [2] 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 low complexity lossless compression for images (LOCO-I) 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.* [7] 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 is avoided. Then, a modified embedded block coding with optimized truncation [4] 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.

The compression method proposed by Faramarzpour *et al.* [8] starts by locating and extracting the microarray spots, isolating each spot into an individual region of interest (ROI). To each of these ROIs, 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.

More recently, Lonardi *et al.* [9] proposed lossless and lossy compression algorithms for microarray images (MicroZip). The method uses a fully automatic gridding procedure, similar to that of Faramarzpour's method, for separating spots from the background (which can be lossy compressed). Through segmentation, the image is split into two channels: foreground and background. Then, for entropy coding, each channel is divided into two 8-bit subchannels and arithmetic encoded, with the option of being previously processed by a Burrows-Wheeler transform.

## III. Standard Methods

JBIG [5], JPEG-LS [4], [6], and JPEG2000 [3], [4] 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. This diversity in coding engines might be helpful when drawing conclusions regarding the appropriateness of each of these technologies for the case of microarray image compression.

### A. 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.* [2]); 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], 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 for 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 I shows the compression results, in 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 bpp). 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.

The average results by image set show that JPEG-LS provides the highest compression in the case of the Apo AI and MicroZip images, whereas JBIG gives the best results for the ISREC set. Lossless JPEG2000 is always slightly behind these two. It is interesting to note that the set for which JBIG gave the best results is also the one requiring more bpp for encoding.

TABLE I
COMPRESSION RESULTS, IN bpp, USING LOSSLESS JPEG2000, JBIG AND JPEG-LS. FOR REFERENCE, RESULTS ARE ALSO GIVEN FOR THE POPULAR COMPRESSION TOOL GZIP

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

### B. Sensitivity to Noise

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 [2]. Since this fact may result in some degradation in the compression performance of the encoders, we decided to address this problem and to study the effect of noisy bit-planes in the compression performance of the standards.

To perform this evaluation, we separated the images into a number $p$ of most significant bit-planes and $16 - p$ least significant bit-planes. Whereas the $p$ most significant bit-planes have been sent to the encoder, the $16 - p$ least significant bit-planes have been left uncompressed. This means that the bitrate of a given image results from the sum of the bitrate generated by encoding the $p$ most significant bit-planes plus the $16 - p$ bits concerning the bit-planes that have been left uncompressed.

Table II compares average results for the three set of images regarding three situations: 1) full compression, i.e., all 16 bit-planes are encoded;2) the image is divided into the eight most significant bit-planes (which are encoded) and the eight least significant bit-planes (which are left uncompressed); 3) the optimum value of $p$ is determined for each image. From Table II we can see that, in fact, this splitting operation can provide some additional compression gains. The best results attained provided improvements of 3.1%, 2.6% and 1.9%, respectively, for JBIG, lossless JPEG2000 and JPEG-LS.

However, finding the right value for $p$ may require as many as 16 iterations of the compression phase, in order to find it. Moreover, from

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

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

[3] The original web-site 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 II
AVERAGE COMPRESSION RESULTS, IN bpp, WHEN A NUMBER OF BIT-PLANES IS LEFT UNCOMPRESSED. THE COLUMNS LABELED "8 PLANES" PROVIDE RESULTS FOR THE CASE WHERE ONLY THE 8 MOST SIGNIFICANT BIT-PLANES HAVE BEEN ENCODED AND THE 8 LEAST SIGNIFICANT BIT-PLANES HAVE BEEN LEFT UNCOMPRESSED. THE COLUMN NAMED "BEST" CONTAINS THE RESULTS FOR THE CASE WHERE THE SEPARATION OF MOST AND LEAST SIGNIFICANT BIT-PLANES HAS BEEN OPTIMALLY FOUND

| Image set | JPEG2000 | | | JBIG | | | JPEG-LS | | |
|---|---|---|---|---|---|---|---|---|---|
| | 16 planes | 8 planes | Best | 16 planes | 8 planes | Best | 16 planes | 8 planes | Best |
| Apo AI | 11.063 | 10.940 | 10.790 | 10.851 | 10.510 | 10.507 | 10.608 | 10.523 | 10.433 |
| ISREC | 11.366 | 11.100 | 10.954 | 10.925 | 10.607 | 10.583 | 11.145 | 10.838 | 10.713 |
| MicroZip | 9.515 | 9.918 | 9.321 | 9.297 | 9.506 | 9.030 | 8.974 | 9.588 | 8.912 |
| **Total Average** | **10.653** | **10.661** | **10.376** | **10.393** | **10.224** | **10.073** | **10.218** | **10.302** | **10.026** |

the results shown in Table II, we can see that a simple separation of the bit-planes in an upper and lower half may improve the compression in some cases (Apo AI and ISREC image sets), but may also produce the opposite result (MicroZip image set).

### C. Lossy-to-Lossless Compression

From the point of view of compression efficiency, and taking into account the results presented in Table I, JPEG-LS is the overall best lossless compression method, followed by JBIG and lossless JPEG2000. The difference between JPEG-LS and lossless JPEG2000 is about 4.1% and between JPEG-LS and JBIG is 1.7%. However, the better compression performance provided by JPEG-LS might be somewhat overshadowed by a potentially important functionality provided by the other two standards, which is progressive, lossy-to-lossless, decoding.

In the case of JPEG2000, this functionality results both from the multi-resolution wavelet technology used in its encoding engine and from a strategy of information encoding based on layers [4]. In the case of JBIG, this property comes from two different sources. On one hand, images with more than one bit-plane are encoded using a bit-plane by bit-plane coding approach. This provides a kind of progressive decoding, from most to least significant bit-planes, where the precision of the pixels is improved for each added bit-plane and the $L_\infty$ error is reduced by a factor of two. On the other hand, JBIG permits the progressive decoding of each bit-plane by progressively increasing its spatial resolution [5]. However, the compression results that we present in Table I do not take into account the additional overhead implied by this encoding mode of JBIG (we used the -q flag of the encoder, which disables this mode).

In Fig. 1, we present rate-distortion curves for image "1230c1G," obtained with the JPEG2000 and JBIG coding standards, and according to two error metrics: norm $L_2$ (root mean squared error) and norm $L_\infty$ (maximum absolute error). Regarding norm $L_2$, we observe that JPEG2000 provides slightly better rate-distortion results for bitrates less than 8 bpp. For higher bitrates, this codec exhibits a sudden degradation of the rate-distortion. We believe that this phenomenon is related to the default parameters used in the encoder, which might not be well suited for images having 16 bpp. Moreover, we think that a careful setting of these parameters may lead to improvements in the rate-distortion of JPEG2000 for bitrates higher than 8 bpp, although we consider this tuning a problem that is beyond the scope of this paper.

With respect to norm $L_\infty$, we observe that JBIG is the one with the best rate-distortion performance. In fact, due to its bit-plane by bit-plane approach, it guarantees an exponential and upper bounded decrease of the maximum absolute error. The upper bound of the error is given by $2^{(16-p)} - 1$, where $p$ is the number of bit-planes already decoded. Contrarily, JPEG2000 cannot guarantee such bound, which may be a major drawback in some cases. Finally, we note that the sudden deviation of the JPEG2000 curves around bitrates of 8 bpp is probably related to the same problem pointed out earlier for the case of the $L_2$ norm.

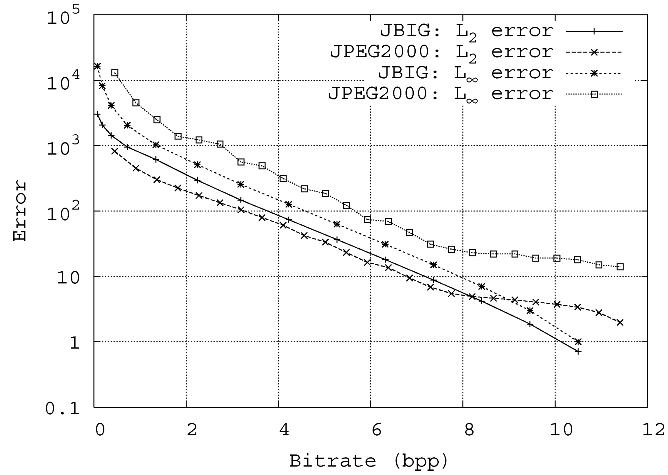


Fig. 1. Rate distortion curves (image "1230c1G") showing the performance of JPEG2000 and JBIG in a lossy-to-lossless mode of operation. Results are given both for the $L_2$ (root mean squared error) and $L_\infty$ (maximum absolute error) norms.

## IV. CONCLUSION

From the experimental results obtained, we conclude that JPEG-LS gives the best lossless compression performance. Moreover, according to the implementations used (not necessarily optimized for speed) it is about four times faster than the other two. However, it lacks lossy-to-lossless capability, which might be a decisive functionality if remote transmission over slow links is a requirement. Regarding the rate-distortion performance, JPEG2000 was the best algorithm according to the $L_2$ error metric, whereas JBIG was the most efficient considering the $L_\infty$ norm. Regarding lossless compression performance, JBIG was consistently better than JPEG2000.

The method that gained most from a correct separation of most significant bit-planes (that are encoded) and least significant bit-planes (that are left uncompressed) was JBIG. It is, simultaneously, the encoding technique that, due to the bit-plane by bit-plane coding, can search for the optimum point of separation more easily. In fact, this can be done by monitoring the bitrate resulting from the compression of each bit-plane, and stopping compressing when this value is over 1 bpp. It also worths mentioning that since JBIG was designed for bi-level images, the bit-planes are compressed independently. Therefore, techniques based on the same technology, but exploiting inter-bit-plane dependencies, most probably will do better. Based on these observations, it is our opinion that the technology behind JBIG seems to be in a good position for attacking the problem of microarray image coding. Moreover, and as demonstrated by the specialized methods already proposed, although the standards play an important role, we have no doubt that the future of microarray image compression depends on special-purpose, dedicated techniques.

## REFERENCES


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

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

[3] *Information Technology—JPEG 2000 Image Coding System*, ISO/IEC Standard 15 444-1, 2000.

[4] D. S. Taubman and M. W. Marcellin, *JPEG 2000: Image Compression Fundamentals, Standards and Practice*. Norwell, MA: Kluwer Academic, 2002.

[5] *Information Technology—Coded Representation of Picture and Audio Information—Progressive Bi-Level Image Compression*, ISO/IEC Standard 11 544, Mar. 1993.

[6] *Information Technology—Lossless and Near-Lossless Compression of Continuous-Tone Still Images*, ISO/IEC Standard 14 495-1, 1999.

[7] J. Hua, Z. Xiong, Q. Wu, and K. Castleman, "Fast segmentation and lossy-to-lossless compression of DNA microarray images," presented at the *Workshop Genomic Signal Processing and Statistics, GENSIPS*, Raleigh, NC, Oct. 2002.

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

[9] S. Lonardi and Y. Luo, "Gridding and compression of microarray images," presented at the *IEEE Computer Society Bioinformatics Conf., (CSB-2004)*, Stanford, CA, Aug. 2004.


# Motion Artifact Reduction in Photoplethysmography Using Independent Component Analysis

Byung S. Kim and Sun K. Yoo*

***Abstract*—Removing the motion artifacts from measured photoplethysmography (PPG) signals is one of the important issues to be tackled for the accurate measurement of arterial oxygen saturation during movement. In this paper, the motion artifacts were reduced by exploiting the quasi-periodicity of the PPG signal and the independence between the PPG and the motion artifact signals. The combination of independent component analysis and block interleaving with low-pass filtering can reduce the motion artifacts under the condition of general dual-wavelength measurement. Experiments with synthetic and real data were performed to demonstrate the efficacy of the proposed algorithm.**

***Index Terms*—Block interleaving, ICA, motion artifact, photoplethysmography.**

## I. INTRODUCTION

Photoplethysmography (PPG) is an electro-optic technique to measure the pulse wave of blood vessels. In pulse oximeter, the measuring apparatus for PPG [1], motion artifacts can limit the accuracy of the measured PPG signal during movement. Particularly, the motion artifacts cannot be easily managed because of the frequency overlapping between PPG and the motion artifact signals [2]. Since general frequency domain filtering methods can be unsuccessful, some methods have been researched to manage the motion artifacts from measured PPG signals [1], [2]. However, further research is still required to improve the performance of motion artifact rejection.

In this paper, the new motion artifact reduction method was proposed under the constraint of dual-wavelength measurement. We combined independent component analysis (ICA) and a signal enhancement preprocessor to separate the PPG signal from the motion artifact-contaminated measured signals. Experiments with synthetic and real data were performed to demonstrate the efficacy of the proposed algorithm.

Manuscript received November 12, 2003; revised July 1, 2005. This work was supported by the Korea Health 21 R & D Project, Ministry of Health & Welfare, Republic of Korea under Grant 02-PJ3-PG6-EV08-0001. *Asterisk indicates corresponding author.*

B. S. Kim is with Graduate School of Biomedical Engineering, Yonsei university, Seoul 120–752, Korea.

*S. K. Yoo is with the Department of Medical Engineering, Center for Emergency Medical Informatics, Human Identification Research Center, Yonsei University College of Medicine, Seoul 120–752, Korea (e-mail: sunkyoo@yumc.yonsei.ac.kr).

Digital Object Identifier 10.1109/TBME.2005.869784

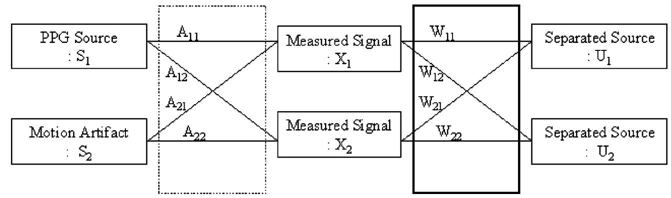


Fig. 1. ICA model for motion artifact separation.

## II. MOTION ARTIFACT REDUCTION

The motion artifact reduction method, consisting of the preprocessor and the ICA, is newly designed based on the quasi-periodicity of PPG signal and the independence between the PPG and the motion artifact signals. The preprocessor enhances the PPG component from measured signal and then the ICA separates the PPG signal from preprocessed signal. The preprocessor consists of period detection, block interleaving, low-pass filtering, and block de-interleaving. In particular, the ICA model with two independent sources is considered to complement the popular dual-wavelength optical probe.

### A. ICA Model for Motion Artifact Separation

The PPG and motion artifact signal sources can be assumed to be independent of each other, since the heart pulsation for the PPG signal has little correlation with the physical movement for the motion artifact signal. As shown in Fig. 1, two measured signals $(\mathbf{X})$, can be modeled as the linear mixture of motion artifact and PPG signal sources $(\mathbf{S})$ with an unknown mixing matrix $(\mathbf{A})$, if they are independent

$$\mathbf{X} = \mathbf{A}\mathbf{S}. \tag{1}$$

The unknown $\mathbf{A}$ and the unknown $\mathbf{S}$ can be estimated from the measured $\mathbf{X}$ (motion artifact contaminated signals) by ICA. The separated sources $\mathbf{U}\ (= \mathbf{S})$, the PPG signal and the motion artifact signal, can be obtained by estimated $\mathbf{W}\ (= \mathbf{A}^{-1})$. The $\mathbf{W}$ can be estimated by a fast ICA algorithm [3], [4]. In other words, the PPG source separation achieves the motion artifact reduction.

However, the actual number of independent sources contained in the measured $\mathbf{X}$ can be more than two. The motion artifact signal is postulated as the complex combination of multiple sources [2]. In addition to the motion artifacts, other noise can be added to $\mathbf{X}$ [1]. In order to separate PPG from multiple sources using the ICA model for two independent sources, the preprocessor should be employed to suppress noise in measured $\mathbf{X}$, which in turn enhances the PPG signal comparing with other noise sources, before applying the ICA model.

### B. Preprocessor for PPG Signal Enhancement

In order to remove noise without the deterioration of the PPG signal, we exploited the quasi-periodicity of the PPG signal. The PPG signal