LOSSLESS AND NEAR-LOSSLESS COMPRESSION OF ECG SIGNALS

Ziya Arnavut
Department of Mathematics and Computer Science
SUNY Fredonia, Fredonia, NY 14063
email: ziya.arnavut@fredonia.edu

Abstract – **In this paper we present Linear Transformation Algorithm (LTA), which is based on a new transformation, Linear Block Transformation (LOT). Experimental results show that Linear Transformation Algorithm yields comparable results to Burrows-Wheeler Algorithm (BWA) [4] and outperforms Gzip, and Shorten Waveform Coder for near-lossless ECG compression; for lossless ECG compression it yields better compression than all the other techniques.**

Keywords: ECG Compression, Lossless, Near-lossless

I. INTRODUCTION

Effective compression of electrocardiogram (ECG) signals is required in many applications including: (a) ECG data storage, (b) ambulatory recording system; and (c) ECG data transmission over the network.

Although lossy compression yields significantly higher compression ratios while preserving diagnostic accuracy, due to legal concerns it is not usually employed. Therefore, we focus on lossless and near-lossless compression of ECG signals.

Various researchers [8], [9] and [10] have investigated the transform-based compression techniques for ECG data. However, Block Sorting techniques have not been fully investigated.

One of the recent developments in the text compression area is the *Block Sorting Lossless Data Compression Algorithm* (BWA) introduced by Burrows and Wheeler [4]. When applied to text or image data, BWA achieves better compression rates than Ziv-Lempel techniques with comparable speed, while its compression performance is close to context-based methods, such as PPM. The lexical sorting transformation utilized in BWA is called Burrows-Wheeler Transformation (BWT).

Clearly et al. [5] viewed BWA (called BW94 by the authors) as a context based method, with no predetermined upper bound to context length. Fenwick [6], has done a comparative study on BWA. He concluded that BWA is a "viable text compression technique, with a compression approaching that of the currently best compressors while being much faster than many other compressors of comparable performance" [6]. Arnavut and Magliveras [1] have generalized the idea for permutations and introduced the *Lexical Permutation Sorting algorithm* (LPSA). They have shown that the BWT is reducible to LPSA, and LPSA has some choices not available with BWA, when the underlying data to be transmitted is a permutation.

Since the introduction of BWA, block-sorting schemes have attracted great attention in compression community. In this work we introduce a different block transformation, *Linear Order Transformation* (LOT), and show that the LOT transformation is faster than the BWT transformation and yields better compression than the BWA, Gzip and Shorten Waveform coder for ECG data.

II. LINEAR ORDER TRANSFORMATION

In this work we assume knowledge of some basic mathematical concepts, including familiarity with elementary properties of standard objects of discrete mathematics.

For the definitions of *permutations* and *multiset permutations* interested reader should consult to [10]. Given a multiset permutation (data string) ω = [3,1,3,1,2], we construct a matrix, *M,* by taking consecutive cyclic left-shifts of ω as the rows of *M*:

$$M = \begin{bmatrix} 3 & 1 & 3 & 1 & 2 \\ 1 & 3 & 1 & 2 & 3 \\ 1 & 2 & 3 & 1 & 3 \\ 1 & 2 & 3 & 1 & 3 \\ 2 & 3 & 1 & 3 & 1 \end{bmatrix}$$

By sorting the rows of *M* lexically we transform it to, *M'*

$$M' = \begin{bmatrix} 1 & 2 & 3 & 1 & 3 \\ 1 & 3 & 1 & 2 & 3 \\ 2 & 3 & 1 & 3 & 1 \\ 3 & 1 & 2 & 3 & 1 \\ 3 & 1 & 3 & 1 & 2 \end{bmatrix}$$

while by sorting the rows of *M* linearly (with respect to the first element in each row), we transform it to

Report Documentation Page

Report Date	Report Type	Dates Covered (from... to)
25 Oct 2001	N/A	-

Title and Subtitle Lossless and Near-Lossless Compression of ECG Signals	**Contract Number**
	Grant Number
	Program Element Number
Author(s)	**Project Number**
	Task Number
	Work Unit Number
Performing Organization Name(s) and Address(es) Department of Mathematics and Computer Science SUNY Fredonia Fredonia, NY 14063	**Performing Organization Report Number**
Sponsoring/Monitoring Agency Name(s) and Address(es) US Army Research, Development & Standardization Group (UK) PSC 803 Box 15 FPO AE 09499-1500	**Sponsor/Monitor’s Acronym(s)**
	Sponsor/Monitor’s Report Number(s)
Distribution/Availability Statement Approved for public release, distribution unlimited	
Supplementary Notes Papers from 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, October 25-28, 2001, held in Istanbul, Turkey. See also ADM001351 for entire conference on cd-rom.	
Abstract	
Subject Terms	
Report Classification unclassified	**Classification of this page** unclassified
Classification of Abstract unclassified	**Limitation of Abstract** UU
Number of Pages 4	

$$M'' = \begin{bmatrix} 1 & 3 & 1 & 2 & 3 \\ 1 & 2 & 3 & 1 & 3 \\ 2 & 3 & 1 & 3 & 1 \\ 3 & 1 & 3 & 1 & 2 \\ 3 & 1 & 2 & 3 & 1 \end{bmatrix}$$

Hence, we obtain two distinct matrices, M' and M'' with respect to two different orderings. M' and M'' have the same rows, but the ordering of the rows is different. Let the first column of M' *be* denoted by F', second column of M' be denoted by S', and the last column of M' *be* denoted by L'. Similarly, let the first column of M'' be denoted by F'', second column of M'' be denoted by S'', and the last column of M'' *be* denoted by L''. Notice that F' and F'' *are* sorted values of ω in ascending order, and other columns are not.

Burrows and Wheeler [4] showed that for any given L' (the last column of M') and the index of the original multiset permutation ω in M', we can recover ω. We now introduce the Linear Ordering Transformation technique and show that it is faster than the Burrows-Wheeler Transformation.

Definition 2.1 *The linearly ordered matrix, $M'' = M''^{m}$, of a multiset permutation* ω *from an underlying set $X = \{ 1, 2, ..., m \}$, is defined as follows:*

1. *Let initially M''^{0}* = [ ] .

2. $M''^{v} = [M''^{v-1}] * [T_v]$, where $v \in X$, $T_v = R_1 * R_2 * \ldots * R_v$, *and a*b denotes appending row b to row a; R_j is the multiset permutation formed by cyclically left-shifting* ω *in (k-1) positions, and k is the position (address) of j^{th} - occurrence of symbol v in* ω..

To clarify the definition, we give an example. Let ω = [3,1,3,1,2] be a multiset permutation from the set $X = \{1,2,3\}$. Initially, let M''^{0} = [ ] be empty. The 1's in ω occur in positions two and four respectively. Since ω has two v = 1, $M''^{1} = [M''^{0}] * [T_1] = [T_1]$ and $T_1 = R_1 * R_2$. By cyclically left-shifting ω in (2-1) = 1 position, we obtain R_1 = [1,3,1,2,3]; while by cyclically left-shifting ω in (4-1) = 3 positions, we acquire R_2 = [1,2,3,1,3]. There is only one 2 in ω and it occurs in the fifth position. Cyclically left-shifting ω in (5-1) = 4 positions, we get $T_2 = R_1$ = [2,3,1,3,1]. Thus $M''^{2} = [M''^{1}] * [T_2]$. There are two 3's in ω and they occur at positions one and three respectively. Hence, $T_3 = R_1 * R_2$, and by cyclically left-shifting ω in (1-1) = 0 position we obtain R_1 = [3,1,3,1,2], and by cyclically left-shifting ω in (3-1) = 2 positions we obtain R_2 = [3,1,2,3,1]. Therefore, $M'' = M''^{3} = [M''^{2}] * [T_3]$.

An obvious observation about the linearly ordered matrix M'' is this: For any two given two-tuples $(F''_i\ S''_i)$ and (F''_k, S''_k) from the first two columns of M'', where $F''_i = F''_k$, then the pair (F''_i, S''_i) appears earlier than the pair (F''_k, S''_k) in M'' (scanning from top to bottom) if and only if the pair appears earlier in ω (scanning from left to right). Clearly, the linear ordering induces a particular order on the pairs of the elements (F'', S''). Because a particular ordering is induced, we can always recover ω uniquely if the row index of ω in M'' and the second column S'' of M'' are known. In our example, ω = [3,1,3,1,2] occurs at position 5 and the second column is S'' = [3,2,3,1,1]. Assume that both the row index of ω and S'' are transmitted to a receiver. Upon receiving S'', the receiver obtains the frequencies of the elements in S'' by using the *count sort* [6]. Once the frequencies of distinct elements in S'' are known, the receiver constructs F'' = [1,1,2,3,3] and has the first two columns (F'', S'') of M'',

$$\begin{pmatrix} 1 & 1 & 2 & 3 & 3 \\ 3 & 2 & 3 & 1 & 1 \end{pmatrix}^{T}$$

Accessing to the fifth position of (F'', S''), the receiver acquires the first two elements of ω, (F''_5, S''_5) = [3,1]. By marking the fifth entry, the receiver eliminates it from further consideration (from the two-tuple (F'', S'').The receiver should determine what follows $S''_5 = 1$ in ω to find the rest of the elements of ω. To discover the third element of ω, the receiver scans the F'' from top to bottom to determine the first unused (unmarked) entry that has a value 1. In our example, this is the first entry where $F''_1 = S''_5 = 1$. Hence, $S''_1 = 3$ should follow $S''_5 = 1$ in ω. The receiver then eliminates consideration of the first entry from the two-tuple (F'' , S''). Since $S''_1 = 3$ is determined, the process is repeated to get the fourth element of ω. Again, the receiver scans F'' to determine the position of the first unused entry which contains $S''_1 = 3$. By finding the first unused entry which contains the value 3 at position four in F'', the receiver easily discovers that the fourth element of ω is $S''_4 = 1$. Again, this entry is eliminated from further consideration. Finally, to find the fifth element of ω, the receiver scans to find the first unused entry which has value 1 in F''. In our example, because the first entry that has a value 1 is used previously, the second entry is considered, $F''_2 = 1$. Therefore, the last element of ω is S'' = 2, and ω = [3,1,3,1,2].

The transformation described above is called *Linea*r *Order Transformation* (LOT). The LOT transformation requires O(2n) time. Let f_j be the frequency of symbol j in a given data stream. With one pass over a given data, frequencies (f_1, $f_2, \ldots, f_m$) of different symbols can be discovered. Hence, for each different symbol v in the data, a starting pointer (address) P_v is determined. For example, for symbol i , the starting address initially would be $P_i = f_1 + f_2 + \ldots + f_{i-1}$.

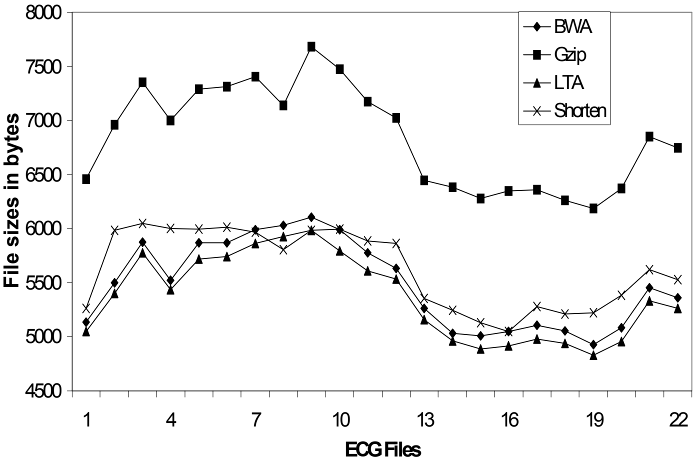

Fig. 1: Lossless compression of ECG files

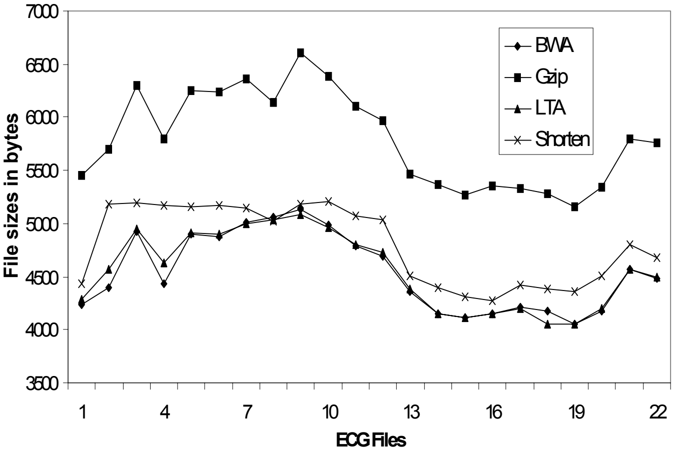

Fig. 2. ECG files with loss value ± 1.

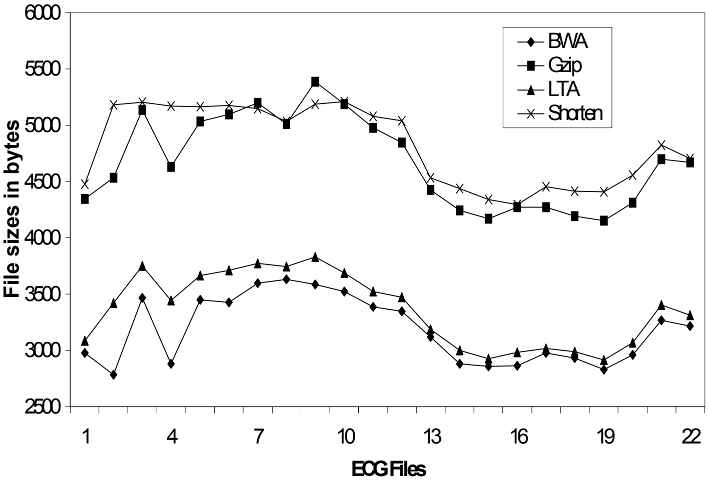

Fig. 3. ECG files with loss value ± 3.

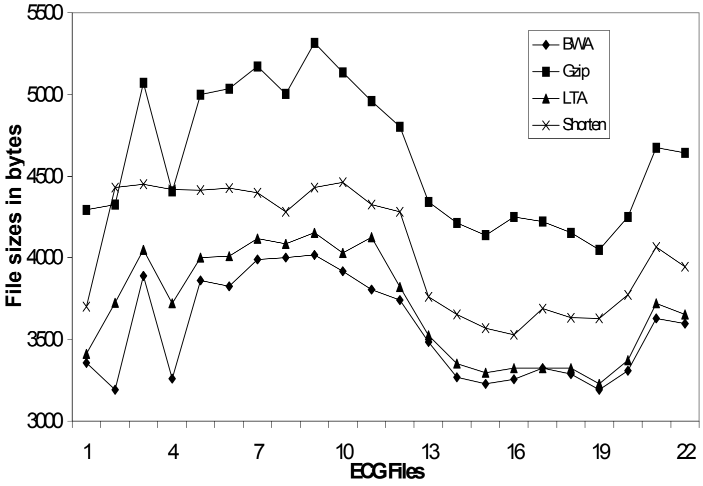

Fig. 4. ECG files with loss value ± 5.

Using those pointers and scanning the data from left to right, for each *v* in the data we write the value of the neighboring element of *v* to the location pointed to by the pointer of *v*. We then update the pointer. Clearly, this operation constructs S*"* in O(2n) time. The time complexity required by the BWT transformation to obtain *L'* is O (n log n) time, because of the lexical sorting [4] . Hence, LOT is faster than BWT. To construct the original data of size n from *S"*, LOT would require O(2n) time, which is also the time required by BWT to construct data from *L'*. When LOT is followed by the MTF [3] and Run-Length coders, we call the technique Linear Transformation Algorithm (LTA), similar to BWA.

III. Experimental results

Figures 1-4 show the compressed sizes of 22 different ECG files with four different techniques. All the files employed in this experiment are obtained from Prof. Memon. Each ECG file is of size 12000 bytes and each ECG signal in the file is recorded with 10-bits. Since each ECG signals is recorded with 10-bits, the BWA and LAT algorithms are modified accordingly. However, the Gzip and Shorten Wave Coder are obtained from public sites and are used without any modifications.

Figure 1 indicates that LTA scheme yields best compression gain out of the three techniques, BWA, Gzip and Shorten, when the files are compressed without any loss. Observe that, when the data files to be compressed have loss value ± 1 (Figure 2) the LTA and BWA yields almost the same compression gain. As can be seen in Figure 3 and 4, when the degree of lost increases, BWA algorithm performs better than the other techniques.

IV. Conclusion

In this work, by expanding the theoretical foundations of *Lexical Permutation Sorting Algorithm* [1] we introduced a new blocks-sorting technique, *Linear Order Transformation*, and showed that LOT is faster than the BWT transformation .

We have shown that when one transforms the data with the LOT transformation followed by the MTF and Run-Length coding, the compression gain obtained is better than the recently introduced BWA and the other well-known compression schemes, such as Gzip and Shorten Waveform coder for lossless ECG data. We also have shown that the Block Transformed data yields better compression than Gzip and Shorten Waveform coder in lossless and near-lossless cases.

Our future work will involve with extending this work to compare the results of all the transformed based coding schemes, such as the ones reported in [8][9][10]. However, considering the results of Gzip as a base for judgment, we believe that Block Sorting Transforms yields better compression gain than the other transform based coders for ECG data.

Acknowledgment

I am thankful to Prof. Memon for supplying me with the ECG data in this work.

References

[1] Arnavut, Z, and Plaku, E., "Lossless Compression of ECG Signals", *Proceedings of the first Joint BMES/EMBS Conference*, P.274, Atlanta Georgia, October 13-16, 1999.

[2] Arnavut, Z., and Magliveras, S. S., "Lexical Permutation Sorting Algorithm" , *The Computer Journal* , Vol. 40, No. 5, Dec. 1997.

[3] Bentley J. L., Sleator, D. D., Tarjan, R. E., and Wei, V. K., "A Locally Adaptive Data Compression Scheme", Co*mmunications of the ACM*, Vol. 29, No. 29, April 1986, pp. 320-330.

[4] Burrows, M. and Wheeler, D. J., "A Block Sorting Lossless Data Compression Algorithm", *SRC Research Report 124*, Digital Systems Research Center, Palo Alto, CA., May 1994. (*ftp site:gatekeeper.dec.com* , */pub/DEC/SRC/research-reports/SRC-124.ps.Z)*

[5] Cleary, J. G., Teahan, W. J. and Witten, I. H., "Unbounded Length Contexts for PPM", *Data Compression Conference*, DCC-95, Snowbird Utah, pp. 52-61, Mar. 1995.

[6] Cormen, T. H., Leiserson, C. E., Riverst, R. L., *Introduction to Algorithms* , McGraw-Hill, 1986.

[7] Fenwick, P., "Improvements over Block-Sorting Text Compression", *The Computer Journal*, Vol. 40, No. 5, Dec. 1997.

[8] Kuklinski, W. S., " Fast Walsh transform data compression algorithm: E.C.G. applications", *Med. Biol. Eng. Comput.*, vol. 21, pp.465-472, July 1983.

[9] Ramakrishnan, A. G. and Saha, S., "ECG Coding by Wavelet-Based Linear Prediction", *IEEE Trans. Biomed.Eng.*, vol. 44, no. 12, Dec. 1997.

[10] Reddy, B. R. S. and Murthy, I.S.N., " ECG data compression using Fourier descriptors", *IEEE Trans. Biomed. Eng.*, vol. BME-33, No. 4, pp. 428-434, 1986.

[11] Stanton, D. and White, D., *Constructive Combinatorics*, Springer-Verlag, 1986.