

High-performance DTW-based signals comparison for the brain electroencephalograms analysis

A.I. Makarova¹, V.V. Sulimova¹

¹Tula State University, Lenin Ave., 92, 300012, Tula, Russia

Abstract

Automatic analysis of electroencephalograms (EEGs) is one of the promising areas of research, which results can be used, in particular, to build systems of mental control of objects. The Dynamic Time Warping procedure (DTW) is used in this work for comparing signals representing EEG. An important feature of the problem we are considering is the need for multiple comparison of signals at the stage of machine learning, which requires enormous computational costs. We propose a parallel algorithm that was implemented in C++ using the MPI technology and tested using the resources of the supercomputer complex of Moscow State University “Lomonosov”. The results of its testing on real data showed that the proposed method allows achieving an almost linear speedup and reducing the total calculation time from 29 days to 3.5 hours using 128 processes, which opens the possibility of improving the quality of automatic analysis of electroencephalograms.

Keywords: high performance computing; comparing fragments of electroencephalograms; Dynamic Time Warping; the MPI technology; the supercomputer complex of MSU “Lomonosov”

1. Introduction

Electroencephalography is a method, which consists in reading electrical signals using electrodes located on the scalp. These signals are generated by the brain in the process of brain activity [1,2] and recorded as electroencephalograms for their subsequent analysis.

Automatic analysis of brain electroencephalograms (EEGs) is one of the promising areas of scientific research, which results can be used, in particular, to build systems for mental control of external devices, e.g. a computer (brain-computer interface, BCI) [3].

The idea of such “brain-computer” interface is based on the fact that signals generated by the brain under the influence of certain (target) stimuli, corresponding to a specific task, contain specific components which are absent under the influence of other stimuli. This happens in particular in the recognition of an object of a given type among a number of other objects. It is obvious that the problem of automatic recognition of the target object on the basis of the analysis of electroencephalograms plays one of the key roles in the construction of a system for the mental control of external objects [4,5]. Thus, it is especially important to improve the quality of its solution.

In this paper we, following [5], focus on the experimental research scheme, according to which a person is provided with a series of mammogram images and his task is to find among them mammograms containing pathologies (the so-called target mammograms). EEG signals from 66 electrodes fixed on different parts of one's head are recorded during the process of viewing images. The task in this case is the automatic detection of signals registered in the process of viewing target mammograms. Figure 1a shows the process of registering electroencephalograms, and figures 1b and 1c - examples of mammograms with pathology and without pathology, respectively [5].

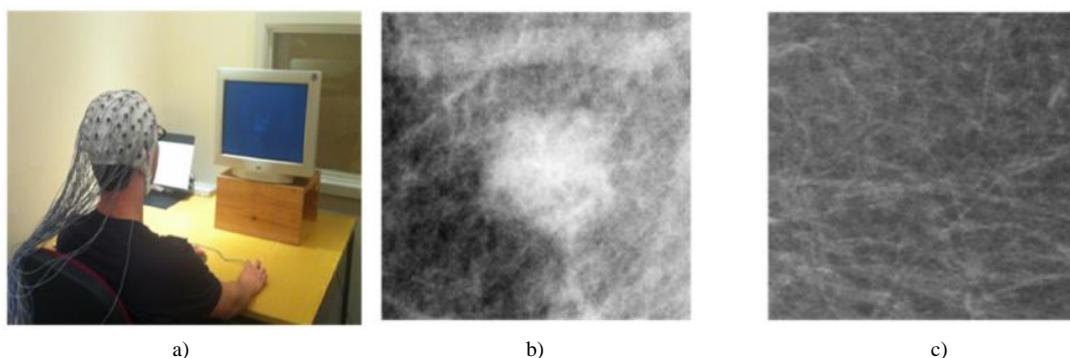


Fig. 1. a) The process of registering electroencephalograms, b) an example of a target object (mammograms with pathology), c) an example of an un-target object (mammograms without pathologies).

Figure 2 shows examples of signals recorded from two of the 66 electrodes during viewing target (solid lines) and non-target (dashed lines) objects. The signals are subjected to preprocessing (filtration, scaling and smoothing by a sliding window), used by us to improve the quality of target objects recognition.

From a mathematical point of view, the electroencephalograms, which need to be analyzed for each of the 66 electrodes, fixed in different parts of the head are single-component discrete signals.

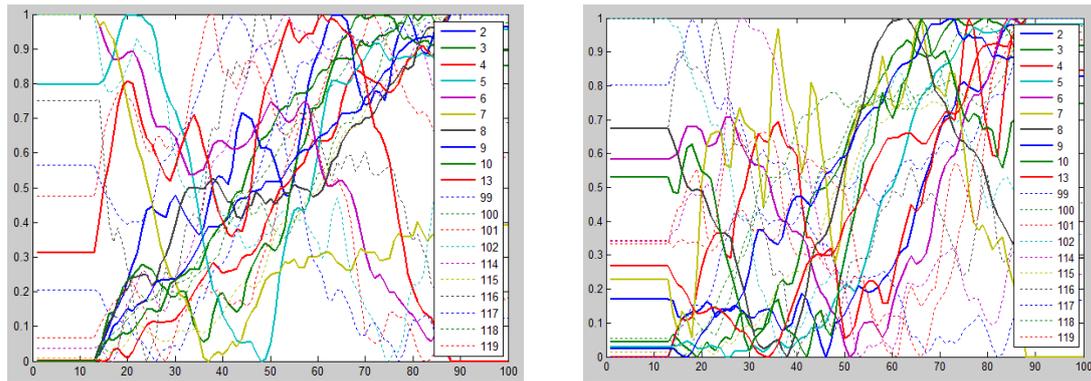


Fig. 2. Examples of signals received from two electrodes. The solid lines show the signals corresponding to the target objects, and the dotted lines show the nontarget objects.

Obviously, the analysis of electroencephalograms must inevitably be based on comparing the signals representing them. Since the form of the response to the stimulus and the time of its onset can vary, in this work, for the comparison of signals an adapted version of Dynamic Time Warping (DTW) procedure is used. This algorithm based on the search for optimal pairwise alignment of compared signals by local compression and stretching of their axes. Initially, the DTW method was proposed to compare speech signals [6], but was later adapted for many other areas [7,8].

The task of analyzing electroencephalograms within the framework of this work is formulated in the form of a two-class pattern recognition problem, which solution is carried out in two stages - training and recognition [9]. At the training stage, a decision rule is formed for assigning new signals to the target or non-target class on the basis of processing a certain finite set of signals with a known class affiliation. At the recognition stage, the constructed decision rule is applied to new signals. Recognition can be carried out very quickly, while the process of constructing a good decision rule, which allows to classify new signals with high accuracy, is very laborious.

Modern methods of machine learning in the construction of decision rules are based on measures of object comparison and allow us to automatically choose the most suitable ones in the training process, improving the quality of the solution of the problem [10]. The great complexity of training is due to the necessity of multiple comparison of long signals in the calculation of a whole series of matrices of their pairwise dissimilarity (for different electrodes, different types of preprocessing, different values of the parameters of the comparison algorithm), choosing the most suitable of which is not possible a priori.

In accordance with the above, an extremely topical task is to increase the productivity of the electroencephalogram comparison.

There are a number of ways to speed up the Dynamic Time Warping procedure, however, they either do not guarantee the finding of the optimal solution [6,11-14], or the performance improvement is provided only in cases, when comparing close or sparse signals [15,16]. But such situations are rare in the analysis of electroencephalograms. The use of modern parallel computing technologies is a fundamentally different direction for increasing productivity of signals comparison. However, the implementation of known parallel versions of DTW procedure (as well as similar tasks with cycles having diagonal dependencies) do not give the desired effect due to the need for frequent synchronization of processes or threads, and in some cases may even lead to an increase in the operating time compared to the serial version because of less efficient work with the cache memory [17-21].

In this paper, we propose a parallel algorithm that significantly improves the performance of calculating the matrix of pairwise comparisons for any signals representing fragments of an electroencephalogram. Increase in productivity is performed without loss of accuracy of calculations due to taking into account the features of the task, allowing to implement parallelization at a higher level.

The proposed algorithm is implemented in C++ programming language using the MPI technology [21] and tested using the resources of the supercomputer complex of Moscow State University "Lomonosov" [23]. The results of the research on real data showed that the proposed method allows to achieve near-linear acceleration, and to reduce the total calculation time from 29 days to 3.5 hours using 128 processes, which opens the possibility of improving the quality of automatic analysis of electroencephalograms.

2. Comparison of fragments of electroencephalograms based on DTW

2.1. The mathematical formulation of the problem of comparing two signals

Let $\mathbf{x} = (x_1, x_2, \dots, x_{N_x})$ and $\mathbf{y} = (y_1, y_2, \dots, y_{N_y})$ - are two single-component discrete signals, which length are N_x and N_y , respectively, and which consists of x_i, y_j OR, $i = 1, \dots, N_x, j = 1, \dots, N_y$ elements.

Specific pairwise warping

$$\mathbf{w}(\mathbf{x}, \mathbf{y}) = \left\{ \begin{matrix} w_{i,1} \\ w_{i,2} \end{matrix}, i = 1, \dots, N_w \right\}, w_{i,1} \in \{1, \dots, N_x\}, w_{i,2} \in \{1, \dots, N_y\}$$

of two signals \mathbf{x} and \mathbf{y} uniquely determines the correspondence of signal's elements and has a length N_w , equal to the number of such pairwise correspondences. At that first and last elements of signals are certainly corresponded to each other:

$$w_{1,1} = w_{1,2} = 1, w_{N_w,1} = N_x, w_{N_w,2} = N_y.$$

A warping of two signals is considered as optimal one if it ensures the minimum value of the following optimality criterion:

$$\hat{\mathbf{w}} = \arg \min_{\mathbf{w}} J(\mathbf{x}, \mathbf{y}, \mathbf{w}), J(\mathbf{x}, \mathbf{y}, \mathbf{w}) = \sqrt{\min_{\mathbf{w}} \sum_{i=1}^{N_x} D(\mathbf{x}, \mathbf{y}, \mathbf{w}, i)}, \quad (1)$$

where

$$D(\mathbf{x}, \mathbf{y}, \mathbf{w}, i) = |x_i - y_i| + \begin{cases} 0, & (w_{i,1} = w_{i-1,1} + 1) \wedge (w_{i,2} = w_{i-1,2} + 1), \\ \beta, & \left\{ \begin{matrix} (w_{i,1} = w_{i-1,1} + 1) \wedge (w_{i,2} = w_{i-1,2}) \\ (w_{i,1} = w_{i-1,1}) \wedge (w_{i,2} = w_{i-1,2} + 1) \end{matrix} \right\} \wedge \wedge u \\ \infty, & \text{иначе.} \end{cases}$$

bi 0 - penalty for non-parallel references between elements of signals, corresponded to local warping axes.

The optimal value of the criterion can be considered as a dissimilarity of the signals:

$$r(\mathbf{x}, \mathbf{y}) = \sqrt{J(\mathbf{x}, \mathbf{y}, \hat{\mathbf{w}})}.$$

2.2. A sequential algorithm for computing the dissimilarity of two signals

The minimum of the optimality criterion (1) can be found by means of the dynamic programming procedure [24]. It is convenient to represent the algorithm of finding the optimal warping in terms of an oriented graph of pairwise correspondences (Figure 3), in which the nodes correspond to the comparison of the signal's elements, the horizontal and vertical edges correspond to the local warping of the axes (passing through them is penalized with the positive penalty β), and the diagonal edges - parallel references between signal's elements.

The algorithm of finding the optimal pairwise warping consists in consecutive passing through all the vertices of the graph, beginning with the upper left and ending with the bottom right vertex. An incomplete value of the optimality criterion $\bar{J}_{i,1}$ is calculated at each vertex on the basis of the initial parts of the signals $\mathbf{x}_{1..i} = (x_1, \dots, x_i)$, $\mathbf{y}_{1..j} = (y_1, \dots, y_j)$:

$$\bar{J}_{1,1} = |x_1 - y_1|,$$

$$\bar{J}_{1,j} = \bar{J}_{1,j-1} + |x_1 - y_j| + \beta, \quad j = 1, \dots, N_y,$$

$$\bar{J}_{i,1} = \bar{J}_{i-1,1} + |x_i - y_1| + \beta, \quad i = 1, \dots, N_x,$$

$$\bar{J}_{i,j} = |x_i - y_j| + \min \{ \bar{J}_{i-1,j-1}, \bar{J}_{i-1,j} + \beta, \bar{J}_{i,j-1} + \beta \}, \quad j = 2, \dots, N_y, i = 2, \dots, N_x.$$

The sought value of the dissimilarity of two signals: $r(\mathbf{x}, \mathbf{y}) = \sqrt{\bar{J}_{N_x, N_y}}$.

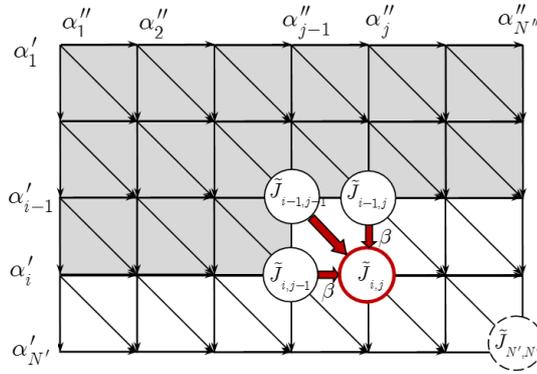


Fig. 3. A graph of pair correspondences of signal samples representing an electroencephalogram.

2.3. Calculation of the matrix of values of dissimilarity of electroencephalograms

As already mentioned above, at the stage of learning the computer algorithms for data analysis, it is necessary to calculate the matrices of the values of dissimilarity for all pairs of signals representing fragments of electroencephalograms from a certain training set $X = \{x_1, x_2, \dots, x_K\}$.

The matrix of pairwise dissimilarities calculated in accordance with the algorithm given in Section 2.2 is symmetric and contains zero values on the main diagonal, so it is sufficient to calculate only the values belonging to the upper (or lower) triangle (Figure 4), the number of which can be determined according to the expression $K(K - 1)/2$.

	x_1	x_2	x_3	...	x_K
x_1	0	$r_{1,2}$	$r_{1,3}$...	$r_{1,K}$
x_2	$r_{1,2}$	0	$r_{2,3}$...	$r_{2,K}$
x_3	$r_{1,3}$	$r_{2,3}$	0	...	$r_{3,K}$
...
x_K	$r_{1,K}$	$r_{2,K}$	$r_{3,K}$...	0

Fig. 4. Matrix of values of pairwise dissimilarity of signals.

Since the number of computations has a quadratic dependence on the number of signals compared, even for a small volume of training set. In this work the training set consists of $K = 755$ fragments of electroencephalograms (for each electrode) and so it is necessary to perform 284635 pairwise comparisons of signals to calculate one matrix. Thus, it is necessary to perform 56357730 pairwise comparisons of signals to calculate such matrices for all 66 electrodes and for three different values of the penalty for warping signal's axes. Since the time of one pairwise comparison requires, on average, 0.045 seconds, the calculation of all the matrices takes about 29 days, which makes it impossible to carry out experiments on real data and requires taking special measures to improve computing performance.

3. Parallel comparison of fragments of electroencephalograms

The task of computing several matrices of pairwise dissimilarity of fragments of electroencephalograms has several levels of data parallelism. Independently from each other can be calculated:

- 1) the incomplete values of the criterion, located on one minor diagonal of the graph of pairwise correspondences,
- 2) all elements belonging to the upper (or lower) triangle of the matrix of values of pairwise dissimilarity (Figure 4),
- 3) all matrices of pairwise dissimilarity.

Parallelization at the first level requires frequent interaction of processes or threads and is associated with the costs of synchronization, what can also lead to inefficient use of the cache [17-21] and, accordingly, does not provide the desired acceleration of computations.

Parallelization at the third level obviously makes sense only if the number of matrices that need to be calculated is greater than the number of available calculators. It also seems inappropriate, since in this study we focus on the computational capabilities provided by the supercomputer complex "Lomonosov" of Moscow State University [23], consisting of more than 5000 nodes and more than 12000 processor cores.

So, in the framework of this paper, parallelization is performed at the second level, i.e. the tasks of calculating the elements of a single matrix of pair-wise dissimilarity for a set of signals are identified as parallel tasks. In this case, if it is necessary to calculate several matrices, then they are computed sequentially.

MPI technology [21] has been chosen as a technology for organizing parallel computing, which allows to organize the interaction of processes running on different computing nodes.

In this case, since even for the Lomonosov supercomputer, the number of elements of each calculated matrix turns out to be much larger than the number of processors, then the parallel tasks are aggregated by means of a one-time distribution of the matrix elements between the processes.

In this paper the following scheme is used to distribute $M = K(K - 1)/2$ elements between the P processes: the first $M \bmod P$ processes receive the $(M - M \bmod P)/P + 1$ elements, and the remaining $M - M \bmod P$ processes receive the $(M - M \bmod P)/P$ elements, where mod is the modulo operation which gives the remainder after division of one integer by another.

This scheme allows us to distribute the work between processes as evenly as possible. The one-time even distribution of elements between the processes proves to be the most effective in this situation, since all compared signals have the same length and, accordingly, the comparison of any pairs of signals takes approximately the same time. As a result, this scheme allows to ensure the most efficient use of the resources of the computer system.

Graphical representation of the scheme of data distribution by processes is shown in Fig. 5.

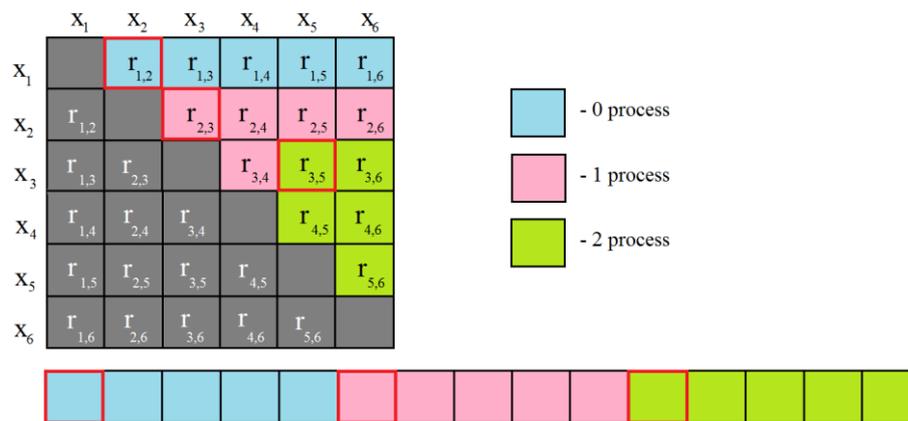


Fig. 5. Scheme of data distribution by processes for three processes.

According to the proposed scheme of parallel computing, each process independently of the others:

- 1) reads the original signals representing fragments of electroencephalograms from the input file,
- 2) determines the number of matrix's elements that it must process and the linear index of the initial element belonging to its range, according to the above-mentioned principle of uniform distribution of elements between processes,
- 3) calculates the row and column number determining the position of the given element in the matrix from the found linear index,
- 4) performs a sequential search of the consecutive elements of the matrix belonging to the upper triangle starting from the element defined in clause 2, at that performs for each element of its range a comparison of the corresponded signals (according to the sequential algorithm described in Section 2.2) and stores the calculated values in their copy of the dissimilarity matrix. Calculations continue until the number of elements obtained in step 2 is processed.

All results are merged on the process with the number 0 using the function MPI_Reduce, after each process computes its elements of the dissimilarity matrix.

4. Experimental study

Brain electroencephalograms obtained in the course of the study described in [5] have been used to investigate the proposed algorithm.

In total, it is required to calculate three matrices (for different values of the penalty) of a pairwise dissimilarity of 755 signals for each of the 66 electrodes. I.e. it is required to calculate 198 matrices. However, the calculation of all matrices of dissimilarity takes approximately the same time and in this connection, it is enough to perform testing for one of the matrices.

The performance study was implemented using the resources of the MSU supercomputer complex “Lomonosov” [23]. Testing was conducted for different matrix sizes and different number of processes to identify possible behavioral features of the proposed algorithm. The results of time measurements in the calculation of one matrix of pair-wise dissimilarity of signals are given in Table 1.

The acceleration, which was obtained for each case, was calculated as the ratio of the sequential computation time to the calculation time on P processes. The results are shown in Table 2.

Table 1. Time of calculating the matrices of pairwise comparison of signals for different matrix sizes and different number of processes

Number of signals	Running time of the algorithm (sec)							
	Number of processes							
	1	2	4	8	16	32	64	128
10	2,14918	1,12535	0,701826	0,48118	0,21083	0,191736	0,205418	0,353755
20	9,04609	4,61894	2,38828	1,15214	0,619012	0,381232	0,297953	0,392734
40	37,1698	18,7533	9,42092	4,82778	2,34974	1,26236	0,822015	0,615929
100	232,34	117,278	58,3369	29,1305	14,696	7,72897	4,09282	2,46599
200	943,653	472,307	238,218	117,96	59,0047	30,2108	15,3994	8,34048
300	2124,3	1083,46	541,713	268,855	133,068	67,1879	34,5807	17,4532
500	5912,42	3010,1	1498,6	736,2	371,859	185,836	93,3412	49,3313
755	13675,7	6864,46	3418,42	1700,52	847,277	422,34	212,281	106,928

Table 2. Acceleration obtained in calculating the matrices of pairwise comparison of signals for different matrix sizes and different number of processes

Number of signals	Acceleration of the algorithm							
	Number of processes							
	1	2	4	8	16	32	64	128
10	1	1,909788	3,062269	4,466478	10,1939	11,20906	10,46247	6,075335
20	1	1,958477	3,787701	7,851554	14,61376	23,72857	30,3608	23,03363
40	1	1,98204	3,945453	7,69915	15,81869	29,44469	45,21791	60,34754
100	1	1,981105	3,982728	7,975833	15,80974	30,06093	56,76771	94,21774
200	1	1,997965	3,9613	7,999771	15,99284	31,23562	61,27856	113,1413
300	1	1,960663	3,921449	7,901285	15,96402	31,6173	61,43022	121,7141
500	1	1,964194	3,945296	8,030997	15,89963	31,81526	63,34202	119,8513
755	1	1,992247	4,000591	8,042069	16,14077	32,38078	64,42263	127,8963

Figure 6 shows graphs illustrating the data from Tables 1 and 2.

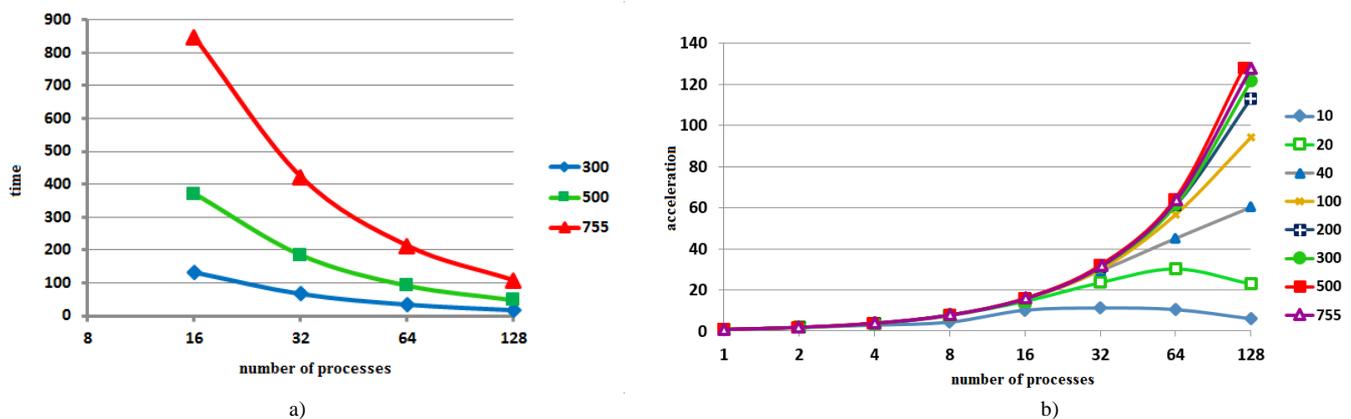


Fig. 6. Graphs of a) time dependence and b) acceleration of the algorithm from the number of processes.

As expected, the acceleration achieved by using the proposed parallel algorithm increases with the size of the calculated matrix of signal dissimilarity. And the acceleration is almost linear in the calculation of the total matrix.

Thus, the proposed approach launched on 128 processes of “Lomonosov” supercomputer complex allowed us to reduce the time of calculating one complete matrix of dissimilarity of fragments of electroencephalograms by 127.89 times. The calculation time for one matrix was reduced from 3.8 hours to 1.78 minutes, and the total calculation time of all 198 matrices was reduced from 29 days to 3.5 hours.

5. Conclusion

In this paper, a high-performance algorithm is proposed which calculates the matrix of the dissimilarity of signals representing fragments of brain electroencephalograms. The proposed algorithm was implemented in C++ programming language with using MPI parallel programming technology. It was tested using the resources of “Lomonosov” supercomputer

complex at Moscow State University [23]. Experimental studies implemented on real data have shown that the proposed algorithm allows to achieve an almost linear speedup and to reduce the total calculation time from 29 days to 3.5 hours by using 128 processes. And so, it opens the possibility of improving the quality of automatic analysis of electroencephalograms.

Acknowledgments

This work is supported by the Russian Fund for Basic Research, grant 15-07-08967.

The authors would like to thank rector of Lomonosov Moscow State University Viktor Sadovnichiy and Moscow State University Supercomputing Center for providing “Lomonosov” supercomputer complex to perform experimental study.

References

- [1] Zenkov LR, Zenkov KS. Clinical electroencephalography (with elements of epileptology). 3rd ed. Moscow: Publishing house MEDPRESS-INFORM, 2004; 368 p. (in Russian)
- [2] Teplan M. Fundamentals of EEG Measurement. *Measurement Science Review* 2002; 2(2): 1–11.
- [3] Wolpaw J, McFarland DJ, Neat GW, Forneris CA. An R. EEG-based brain-computer interface for cursor control. *Electroencephalography & Clinical Neurophysiology* 1991; 8(3): 252–259.
- [4] Tran L. EEG Features for the Detection of Event-Related Potentials Evoked Using Rapid Serial Visual Presentation. PhD Thesis 2014; 63 p.
- [5] Hope C, Sterr A, Langovan PE, Geades N, Windridge D, Young K, Wells K. High Throughput Screening for Mammography using a Human-Computer Interface with Rapid Serial Visual Presentation (RSVP). *Proc. SPIE 8673, Medical Imaging: Image Perception, Observer Performance, and Technology Assessment* 2013; 867303. DOI:10.1117/12.2007557
- [6] Sakoe H, Chiba S. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing* 1978; 26(1): 43–49.
- [7] Berndt DJ, Clifford J. Using dynamic time warping to find patterns in time series. *Association for the Advancement of Artificial Intelligence, Workshop on Knowledge Discovery in Databases* 1994: 229–248.
- [8] Keogh E, Pazzani M. Scaling up dynamic time warping for datamining applications. *Proceedings of the sixth ACM SIGKDD intern. conf. on Knowledge discovery and data mining*. ACM Press, New York, NY, USA 2000: 285–289.
- [9] Vapnik V. *Statistical Learning Theory*. John-Wiley & Sons, Inc., 1998.
- [10] Tatarchuk A, Sulimova V, Mottl V, Windridge D. Supervised Selective Kernel Fusion for Membrane Protein Prediction. *Lecture Notes in Computer Science* 2014; 8626: 98–109.
- [11] Myers C, Rabiner LR, Rosenberg AE. Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing* 1980; 28(6): 623–635.
- [12] Keogh E, Ratanamahatana C. Exact indexing of dynamic time warping. *Knowledge and Information Systems* 2004; 7(3): 358–386.
- [13] Lemire D. Faster retrieval with a two-pass dynamic-time-warping lower bound. *Pattern Recogn.* 2009; 42(9): 2169–2180.
- [14] Salvador S, Chan P. Toward accurate dynamic time wrapping in linear time and space. *Intelligent Data Analysis* 2007; 11(5): 561–580.
- [15] Al-Nayma G, Chawla S, Taheri J. SparseDTW: A Novel Approach to Speed up Dynamic Time Warping, 2012
- [16] Silva DF. Speeding up all-pairwise dynamic time warping matrix calculation 2016. URL: <http://sites.labc.icmc.usp.br/prunedDTW>
- [17] Lamport L. The parallel execution of DO loops. *Commun. ACM* 1974; 17(2): 83–93.
- [18] Babichev AV, Lebedev VG. Parallelization of program cycles. *Programming* 1983; 5: 52–63. (in Russian)
- [19] Fernandez A, Llberia JM, Valero-Garcia M. Loop Transformation Using Nonunimodal Matrices. *IEEE Transactions on Parallel and Distributed Systems* 1995; 6(8): 832–840.
- [20] Abu Khalil JM, Morylev RI, Shteinberg BI. Parallel Algorithm of Global Alignment with Optimal Memory Usage. *Modern problems of science and education* 2013; 1. URL: <http://www.science-education.ru/107-8139>. (in Russian)
- [21] Steinberg BI. Optimizing the use of the memory cache in computational tasks and optimizing compilation. *The All-Russian Scientific Conference on Informatics Problems SPISOK-2013, mekmat of St. Petersburg University, St. Petersburg, 2013*. (in Russian)
- [22] Antonov AS, Tutorial A. *Parallel Programming Using MPI Technology*. Moscow: MGU, 2004; 71 p. (in Russian)
- [23] Voevodin VIV, Zhumatiy SA, Sobolev SI, Antonov AS, Bryzgalov PA, Nikitenko DA, Stefanov KS, Voevodin VadV. *Practice of ‘Lomonosov’ Supercomputer*. Open Systems. Moscow: “Open Systems” Publishing house 2012; 7: 36–39. (in Russian)
- [24] Bellman R, Kalaba RM. *Dynamic Programming and Modern Control Theory*. Science 1969; 118 p. (in Russian)