Computer Science and Information Systems 12(1):49–62 DOI: 10.2298/CSIS140101083B

Fast DCT Algorithms for EEG Data Compression in Embedded Systems

Darius Birvinskas[1], Vacius Jusas[1], Ignas Martisius[1], and Robertas Damasevicius[1]

Software Engineering Department,
Kaunas University of Technology
Studentu St. 50 - 404, Kaunas, Lithuania
darius.birvinskas@ktu.lt

Abstract. Electroencephalography (EEG) is widely used in clinical diagnosis, monitoring and Brain - Computer Interface systems. Usually EEG signals are recorded with several electrodes and transmitted through a communication channel for further processing. In order to decrease communication bandwidth and transmission time in portable or low cost devices, data compression is required. In this paper we consider the use of fast Discrete Cosine Transform (DCT) algorithms for lossy EEG data compression. Using this approach, the signal is partitioned into a set of 8 samples and each set is DCT-transformed. The least-significant transform coefficients are removed before transmission and are filled with zeros before an inverse transform. We conclude that this method can be used in real-time embedded systems, where low computational complexity and high speed is required.

Keywords: Fast DCT, data compression, electroencephalography.

1. Introduction

Electroencephalography (EEG) is a recording of cortically evoked electro-potentials across the scalp. Electrical activity of the scalp evokes oscillations at a variety of frequencies. This oscillatory activity is classified into different frequency bands or rhythms: delta (0.5 - 3.5 Hz), theta (4 - 8 Hz), alpha 1 (8 - 10.5 Hz), alpha 2 (10.5 - 13 Hz) beta 1 (13 - 21 Hz), beta 2 (20 - 32 Hz), and gamma (36 - 44 Hz) [16].

Since EEG discovery by Berger et. al [1], many research activities have centred on extracting useful information about the brains conditions based on the characteristics of the signal. EEG signals are used in clinical diagnosis of brain death, epileptic seizures and other diseases, as well as in intensive care monitoring. Non-clinical uses include neuroscience, psycho-physiological, Brain - Computer Interface (BCI) and other researches. Many applications require acquisition, storage, and automatic processing of EEG during an extended period of time [16].

Since 1970s research on Brain - Computer Interface systems began at the University of California Los Angeles [22], the EEG signal has become the main data source for BCI study due to its low cost and non-invasive nature. It is possible to record different electrical potentials from the central nervous system of a human brain, depending on the subject's mental state. Because EEG signals are non-stationary and non-linear, and normally interfered by eye movements and muscle noises, it is difficult to differentiate the classes of mental tasks so various machine learning algorithms are used for classification.

Moreover, the amount of raw data produced in recording trials is impractical for most machine learning algorithms [2], this can be overcome by transforming time series into feature vectors with a fixed dimensionality [11].

Due to the large data size of the EEG resulting from large number of electrodes, long recording time and usually high sample rate, data compression is required for efficient data transmission and archiving. Efficient compression of the EEG signal is a difficult task due to inherent randomness in the signal, and hence high compression rates cannot be achieved with lossless compression methods [20, 7].

Recently low cost, portable EEG monitoring devices became available, making commercial, non-clinical applications possible [17]. For example, NeuroSky "MindSet" is an audio headset with a single electrode, recording scalp electro-potentials. It uses wireless technology to transmit to a host device. Since "MindSet" is a portable device, long recordings are possible, however working time is limited by battery capacity. In this case lossy compression techniques may be acceptable as long as the reconstructed signal preserves enough information about the user's mental state [14]. For these reasons, simple, low complexity, embedded data compression methods are required, even if it might effect accuracy.

Discrete Cosine Transform (DCT) is a transformation method for converting a time series signal into basic frequency components. An important feature of DCT is that it takes correlated data as input and concentrates it's energy into just the first few transform coefficients. This method is widely used for audio and image compression [19]. Beside that, DCT has been successfully used for EEG data set reduction and feature extraction [2, 3].

DCT and inverse DCT (IDCT) are highly computationally intensive, creating prerequisites for performance bottlenecks in systems utilizing them [15]. To overcome this problem, a number of fast algorithms have been proposed [5, 12, 13] for more efficient computations of these transforms. These algorithms are based on the sparse factorizations of the DCT matrix and many of them are recursive.

In this paper we consider the use of fast Discrete Cosine Transform (fDCT) algorithms for lossy EEG data compression in order to decrease communication bandwidth and transmission time. This method could be useful in portable or low cost devices where data compression is required and small data loss is acceptable.

The rest of the paper is organized as follows. In Section 2, we provide an overview of some recent approaches for EEG compression. We present the method of Discrete Cosine Transform and quality measures of signal compression in Section 3. We discuss the materials and methods used for the experiment in Section 4. We report the results of the experiment in Section 5 and finish with conclusions in Section 6.

2. Related Work

Compression techniques practically aim at obtaining maximum data volume reduction while preserving the significant information for reconstruction. Data compression can be lossless, when the signal waveform fidelity is totally preserved, or lossy, in cases where a certain amount of distortion in the decompressed data is allowed.

2.1. Lossless Compression

High compression rates cannot be achieved with lossless compression methods due to randomness of EEG signal. In some cases conventional compression algorithms, like TAR or ZIP can be used, however as shown in [24] specialized algorithms can achieve higher compression ratio.

A common way to improve compression ratio is the use of mathematical transforms. A method for multi channel EEG compression using Karhunen-Loeve transform (KLT) is shown in [23]. Main disadvantage of KLT is high computation time. In [21], an adaptive error modelling technique for lossless compression has been applied to improve the compression efficiency. Another approch is to use neural network predictors for contex-based error modelling [16]. This has shown some improvement in compression efficiency by removing bias offset of raw data.

2.2. Lossy Compression

Unlike lossless compression, lossy algorithms does not allow perfect reconstruction of the signal, but can achieve higher compression ratio and use less computation resources. Recent works on lossy EEG compression includes the use of Wave Packet Transform [4]. This Cardenas-Barrera et al. proposed method is designed to be low-power and can be used in portable devices. Another approach with Wavelet dictionaries is presented in [8]. The compression is based on a superposition of dictionary elements, by minimizing the norm of the error.

Higgins et al. suggest the use of JPEG2000-based algorithm for lossy EEG compression [9]. The JPEG2000 algorithm is designed for image compression, however applications are not limited to image files only. Core components of algorithm include: Discrete Wavelet Transform (DWT), quantisation and an Arithmetic Coder. The DWT replaces the Discrete Cosine Transform (DCT) of the original JPEG format. Using different compression parameters compression ratio of 11:1 can be achieved [9].

Further, recent works based on wavelet-SPIHT [6] and finite rate of innovation technique [18] has been published. These approaches have shown some improvement in the compression performance at the expense of computation resources. A performance survey of some EEG compression techniques can be found in [24].

Several compression methods have been reported recently, however, to the best of our knowledge none of these approaches uses Discrete Cosine Transform or fast DCT algorithms for EEG data compressions. This method is suitable for hardware implementation and should reduce computation time in embedded systems.

3. Theory

A theoretical background required for our experiment can be divided into to parts. The first one explain Discrete Cosine Transform and some of its properties. Since direct DCT formula implementation is inefficient, three fast DCT computation schemes are also provided. The second part provide information about evaluation parameters that we are going to use in our experiments. These parameters are compression ratio and percent root-mean-square difference (PRD).

3.1. Discrete Cosine Transform

DCT is a transformation method for converting a time series signal into basic frequency components. Low frequency components are concentrated in first coefficients, while high frequency - in the last ones.

The DCT input $f(x)$ is a set of N data values (EEG samples, audio samples, or other data) and the output $Y(u)$ is a set of N Discrete Cosine Transform coefficients. The one-dimensional DCT for a list of N real numbers is expressed by formula (1):

$$Y(u) = \sqrt{\frac{2}{N}} \cdot \alpha(u) \cdot \sum_{x=0}^{N-1} f(x) \cdot cos\left(\frac{\pi \cdot (2x+1) \cdot u}{2N}\right) \tag{1}$$

where

$$\alpha(u) = \begin{cases} \frac{1}{\sqrt{2}},\ u = 0 \\ 1,\ u > 0 \end{cases}$$

The first coefficient $Y(0)$ is called the DC coefficient and rest are referred to as AC coefficients [19]. The DC coefficient contains mean value of original signal.

Inverse DCT takes transform coefficients $Y(u)$ as input and converts them back into time series $f(x)$. For a list of N DCT coefficients inverse transform is expressed by the following formula (2):

$$f(x) = \sqrt{\frac{2}{N}} \cdot \alpha(u) \cdot \sum_{u=0}^{N-1} Y(u) \cdot cos\left(\frac{\pi \cdot (2x+1) \cdot u}{2N}\right) \tag{2}$$

In this formula notations are the same as in (1). DCT exhibits good energy compaction for correlated signals. If the input data consists of correlated quantities, most of the N transform coefficients produced by the DCT are zeros or small numbers, and only a few coefficients are large. These small numbers can be quantized coarsely, usually down to zero. Since EEG has low frequency oscillations, most of the relevant information is compressed into the first coefficients, while the last ones usually contain noise.

During the transform small values of N such as 3, 4, or 6 result in many small sets of data items and small sets of coefficients where the energy of the original signal is concentrated in a few coefficients, but there are not enough small coefficients to quantize. Large values of N result in a few large sets of data. The problem in this case is that individual data items of a large set are normally not correlated and therefore result in a set of transform coefficients where all the coefficients are large. Most data compression methods that employ the DCT use the value of N = 8 [19].

Direct implementation of formulas (1) and (2) requires 4N multiplications, so it is slow and impractical. To overcome this, a number of fast DCT computation algorithms were proposed. These algorithms reduce the number of multiplications and additions, usually, however, limiting the number of samples to 2^m, where m is a positive integer.

Chen DCT Chen et. al [5] published the first fast DCT and IDCT algorithm. They proposed a recursive algorithm to factor any N-point DCT with $N = 2^m, m \geq 2$ into plane rotations and "butterfly" operations. The factorization has a very regular structure and is 6 times faster than the fast Fourier transform based algorithms. The algorithm uses

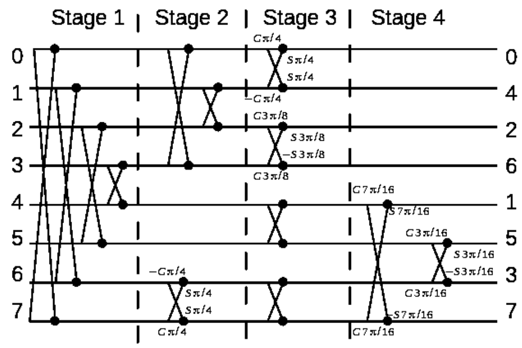

Fig. 1. Signal graph of the Chen DCT transform [5]

$3N/2(log_2N - 1) + 2$ addition operations and $N \cdot log_2N - 3N/2 + 4$ multiplications. For an 8 point vector this number is 16 multiplications and 26 additions. The signal graph for $N = 8$ signal, is presented in Fig. 1.

Loeffler DCT Another factorization for DCT was proposed in [13]. This method uses only 11 multiplications and 29 additions for a 8-point vector, achieving the multiplication lower bound, without an increase in addition operations. One of its variations is adopted by the Independent JPEG Group in a popular image compression algorithm JPEG implementation. The Loeffler Discrete Cosine Transform computation algorithm is sectioned into 4 stages, which are executed in series (see Fig. 2). Note that this factorization requires a uniform scaling factor of $1/\sqrt{8}$ at the end of the flow graph to obtain the true DCT coefficients.

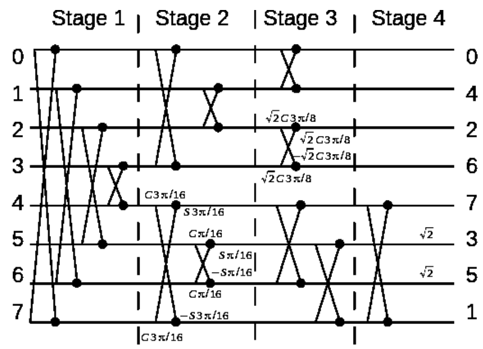

Fig. 2. Signal graph of the Loeffler DCT transform [13]

BinDCT BinDCT, published in [12], is one of the newest fast DCT and IDCT algorithms. Both Chen and Loeffler algorithms need floating point multiplication. While they can be scaled down to integer multiplication, the complexity is still unsatisfactory for embedded systems, requiring a 32 bit data bus. The BinDCT algorithm's novelty lies in that it uses a scheme of lifting steps, which in turn use only shift and addition. This algorithm uses an underlying Chen or Loeffler signal graph, swapping plane rotations and "butterflies" for lifting step operations. These lifting steps are shown in Fig. 3.

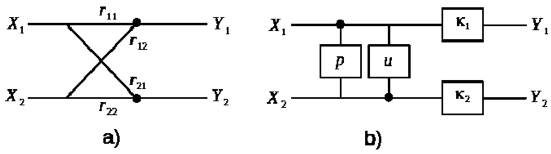

Fig. 3. a) plain rotation, b) lifting steps of the BinDCT algorithm [12]

Lifting step coefficients denoted as p, u, K_1 and K_2 in Fig. 3 b) are calculated according to (3):

$$\begin{aligned} p &= \frac{r_{12}}{r_{11}} \\ u &= \frac{r_{11}r_{21}}{r_{11}r_{22} - r_{21}r_{12}} \\ K_1 &= r_{11} \\ K_2 &= \frac{r_{11}r_{22} - r_{21}r_{12}}{r_{11}} \end{aligned} \tag{3}$$

In this formula coefficients r_{11}, r_{12}, r_{21} and r_{22} are shown in Fig. 3 a). Also in formula (3) all coefficients must have format of $\frac{a}{2^b}$, where a is an integer and b is a natural number. This property is important, because coefficients of this format can be implemented using only logic shifts instead of multiplication and division.

3.2. Evaluation Parameters

Lossless data compression are usually evaluated using compression ratio (CR) [19], which is defined as (4):

$$CR = \frac{L_{comp}}{L_{orig}} \tag{4}$$

where L_{orig} and L_{comp} is the length of original and compressed signals.

However, evaluating CR is not enough for lossy compression. It is well known that higher data compression can be achieved by reducing the quality of reconstructed data. In this paper, we validate the quality of reconstructed signal as a percent of root-mean-square difference (PRD).

Let X and $\tilde{X}$ be the original and the reconstructed signals respectively, and N is its length. The PRD is defined as (5):

$$PRD = \sqrt{\frac{\sum_{n=1}^{N}(x[n] - \tilde{x}[n])^2}{\sum_{n=1}^{N}(x[n])^2}} \cdot 100\% \tag{5}$$

where $x[n]$ and $\tilde{x}[n]$ are samples of original and reconstructed signals. This quantifies the average quality of reconstruction, in contrast to peak signal-to-noise ratio (PSNR), which measures local or worst-case distortion.

Decision between CR and PRD has to be made according to the application. For telemedicine and diagnosing purposes, requiring high accuracy, are opposite to non-clinical uses, where higher compression ratio is preferred.

4. Material and Methods

For experiments two data sets from the BCI competition II (http://bbci.de/competition/ii) were used. These data sets were used in competition and several independent teams have confirmed they are accurate.

The Data set Ia (provided by University of Tuebingen). This data set was recorded using a healthy subject. The subject was asked to move a cursor up and down on a computer screen, while his cortical potentials were taken. During the recording, the subject received visual feedback of his slow cortical potentials (SCPs). Each trial consists of 896 samples from each of 6 channels. Channels locations are these: A1-Cz (10/20 system), A2-Cz, 2 cm frontal of C3, 2 cm parietal of C3, 2 cm frontal of C4, 2 cm parietal of C4.

The Data set IVa (provided by Fraunhofer-FIRST, Intelligent Data Analysis Group and Freie Universitat Berlin, Department of Neurology, Neurophysics Group). This data set was recorded using a healthy subject during a no-feedback session. The task was to press with the index and little fingers the corresponding keys in a self-chosen order and timing. 28 EEG channels were measured and band-pass filter between 0.05 and 200 Hz was applied. Channels are in the following order: F3, F1, Fz, F2, F4, FC5, FC3, FC1, FCz, FC2, FC4, FC6, C5, C3, C1, Cz, C2, C4, C6, CP5, CP3, CP1, CPz, CP2, CP4, CP6, O1, O2.

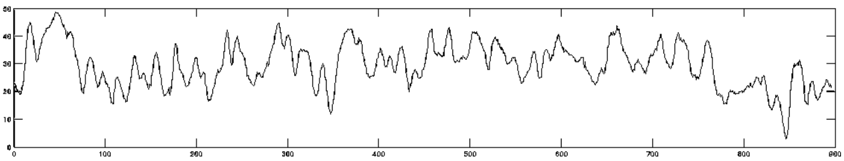

Fig. 4. EEG signal in time series

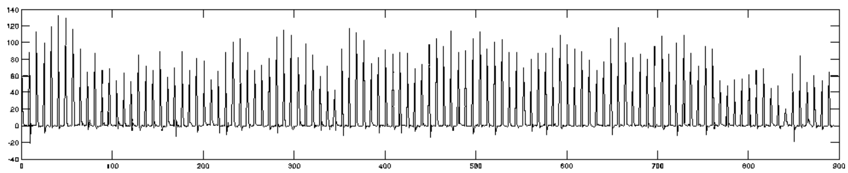

Fig. 5. DCT transform of EEG signal (N=8)

Data sets has several EEG channels, however for our experiment we use only channel 1 from both data sets. Data in other channels are analogous so we assume that compression results would be similar for other channels in the same data set. Since these data sets consist of different number of samples, 200 trials of 896 samples were formed from each data set. The sample count was chosen according to the data structure of Data set Ia. An example of one such EEG trial is shown in Fig. 4. Every trial was compressed and reconstructed separately and average PRD measured. These steps were taken in order to compare experimental results with different data structures, allowing for direct result comparison.

The idea behind this compression algorithm is similar to the one used in audio compression. The EEG signal is partitioned into sets of N samples and each set is then DCT-transformed. As mentioned before, this compresses relevant information into the first DCT coefficients. This step is illustrated in Fig. 5, where N = 8 is chosen. The picture shows the last coefficients being close to zero, making it possible to remove them and therefore reducing the signal length.

In order to speed up computation a number of fast algorithms were proposed. In our experiments the following DCT/IDCT computation algorithms are used:

- Matlab native function.
- Chen's [5] mathematical model using fixed point numbers.
- Loeffler's [13] mathematical model using fixed point numbers.
- BinDCT [12] mathematical model using fixed point numbers.

Matlab native functions use floating point numbers and direct implementation of formulas (1) and (2), so it is expected to be slow, but more accurate. Fast DCT algorithms are implemented in Matlab using fixed point numbers. As a result, fixed point mathematical models are less accurate, but less computationally demanding, and are possible to implement in hardware. Loeffler's mathematical model used in this experiment has already been FPGA-proven and shown to be accurate [15].

There are two important parameters that must be chosen in order to compress a signal. The first one is the signal partition size N. Since fast DCT computation algorithms use a length of 2^m, where m is a positive integer, an obvious choice for partition size is N = 8. This size is used in the most of compression algorithms [19].

Another parameter considered is the number of DCT coefficients to truncate (we denote by R). By removing more coefficients, higher compression ratio is achieved. However, this would reduce the quality of the reconstructed signal. The trade-off between compression ratio and reconstructed signal quality is the subject of our experiment. We chose an R value of 2, 3, 4, 5 and 6. Since compression ratio depends on this value, formula (4) can be rewritten as:

$$CR = \frac{N - R}{N} \tag{6}$$

Our experiments follow these steps:

1. Each trial is partitioned into sets of 8 samples.
2. DCT transform is applied to every set of samples.
3. Number R of last DCT coefficients is removed and a new shorter data vector is constructed.

4. Removed coefficients are filled up with zeros.
5. Inverse DCT transform is applied to every set of coefficients.
6. PRD of each trial is computed.
7. Average PRD is computed.

These steps can be divided in 3 groups: signal compression (steps 1 - 3), signal reconstruction (steps 4 and 5) and quality assessment (steps 6 and 7). Four experiments were carried out using a pair of all previously mentioned DCT/IDCT algorithms (Matlab native function, Chen's, Loeffler's and BinDCT).

5. Results and Discussion

This experiment was carried out using Matlab and a PC running Debian Linux with an Intel Pentium Dual-Core E5700 (3.00 GHz) processor and 6 GB RAM. In order to find out which compression algorithm is faster, we measured time taken to compress and reconstruct the whole data set. In Table 1, relation between compression ratio (CR) and quality of reconstructed signal is shown. This quality is measured using PRD as shown in formula (5).

Table 1. Results using Matlab native DCT function

	Data set Ia		Data set IVa	
CR	PRD, %	time, sec	PRD, %	time, sec
1.33	1.12	4.16	7.92	2.59
1.6	1.41	4.17	8.80	2.54
2	1.74	4.17	9.37	2.54
2.66	2.15	4.18	10.10	2.54
4	4.86	4.17	11.09	2.62

As expected, the signal quality depended on compression ratio, but compression - reconstruction time was constant. We can see that maximum PRD at compression ratio 4 is 4.86% for data set Ia and 11.09% for data set IVa. Cardenas et.al [4] suggest that 99.5% of EEG signal information is maintained when PRD is below 7%. However, other studies suggest that maximum tolerable level for an automated epileptic seizure detection is 30% [9, 10].

Differences between results of different data sets can be explained by their properties. In this case peak-to-peak amplitude is the most notable signal property that differs. While the amplitude of data set Ia is below 100, data set IVa is over 1500. Higher amplitude may reduce correlation between adjacent samples.

In Table 2, Chen's DCT algorithm performance is presented.

We can see that compression time is significantly reduced, but reconstruction quality is lowered as well. For data set Ia quality difference between Chen's and Matlab native algorithms is higher for low CR, but in any case, PRD differs less than 1%. On the other hand, compression - reconstruction time is decreased about 3 times.

Reconstruction quality of data set IVa is significantly lower, compared to Matlab native function results. As mentioned before, models of fast DCT algorithms are designed

Table 2. Results using Chen DCT

	Data set Ia		Data set IVa	
CR	PRD, %	time, sec	PRD, %	time, sec
1.33	2.02	1.47	12.90	0.92
1.6	2.16	1.48	13.51	0.91
2	2.34	1.46	13.89	0.91
2.66	2.63	1.45	14.46	0.92
4	5.07	1.48	15.17	0.93

Table 3. Results using Loeffler DCT

	Data set Ia		Data set IVa	
CR	PRD, %	time, sec	PRD, %	time, sec
1.33	2.70	1.62	7.92	1.02
1.6	2.59	1.58	8.80	1.01
2	2.77	1.57	9.37	0.99
2.66	2.79	1.58	10.10	0.99
4	5.13	1.58	11.09	1.00

for fixed-point hardware implementation, so there is a fixed-point data type overflow possibility.

Table 3 shows results using Loeffler's algorithm. In this case PRD of data set IVa is similar to Matlab's native functions, so we conclude that Loeffler's fast DCT model performs well for both data sets. However, reconstruction quality of data set Ia is slightly lower than Chen's algorithm.

Comparing Table 2 and Table 3 we can see that Loeffler's algorithms is slower, but this difference is low and insignificant. On the other hand reconstruction quality of the dataset IVa is about 5 % higher using Loeffler's algorithm as opposed to Chen's.

In Table 4, results using BinDCT algorithm are presented. This algorithm is faster, but less accurate than Loeffler's.

Table 4. Results with BinDCT

	Data set Ia		Data set IVa	
CR	PRD, %	time, sec	PRD, %	time, sec
1.33	2.68	1.44	8.15	0.90
1.6	2.96	1.42	9.19	0.90
2	3.23	1.42	9.72	0.89
2.66	3.41	1.44	10.33	0.90
4	5.49	1.43	11.08	0.94

Table 3 and Table 4 shows that multiplierless BinDCT algorithm performs similar as Loeffler's algorithm. This can be useful, when designing low-cost, low power embedded compression systems.

A summarized quality result of all the algorithms is shown in Fig. 6 for data set Ia and Fig. 7 for data set IVa. Despite Chen's algorithm overflow with data set IVa, experimental

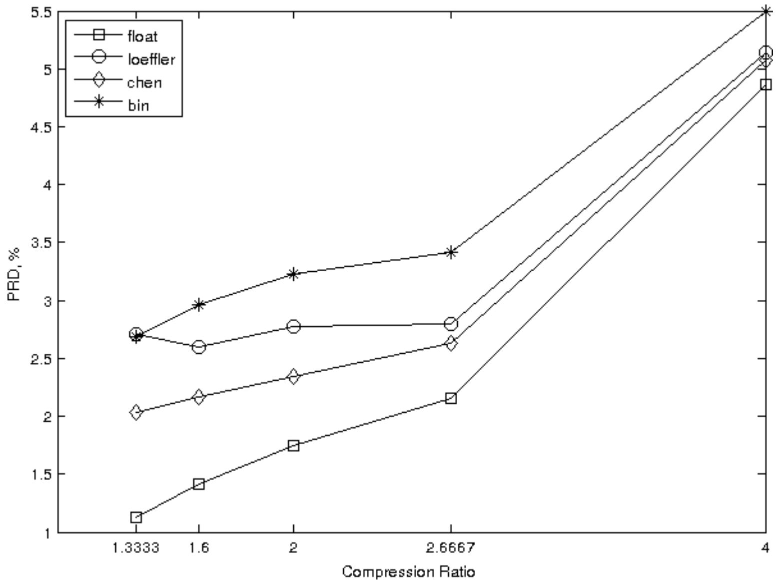

Fig. 6. Quality of reconstructed signal (Data set Ia)

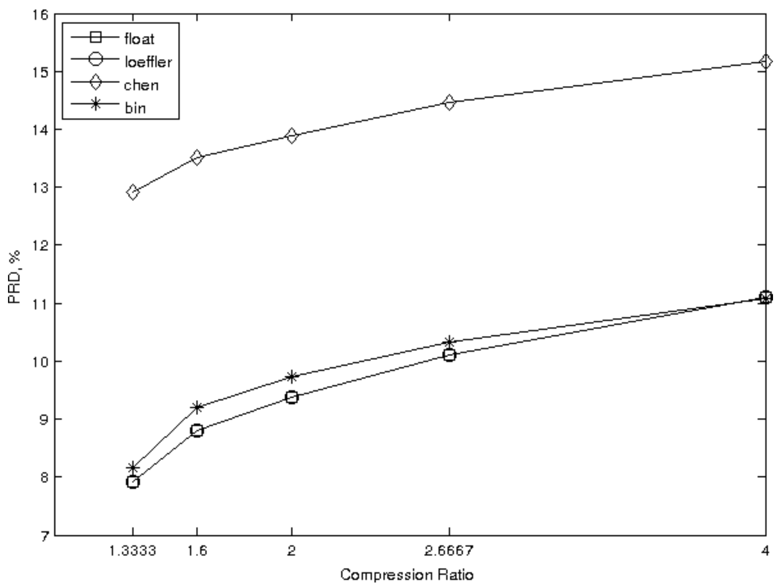

Fig. 7. Quality of reconstructed signal (Data set IVa)

results show that the quality of a reconstructed signal is reduced less than 1%, while using fixed-point fast DCT algorithms. On the other hand, compression - reconstruction time is reduced by about 300%.

Comparing our results to [9] shows that higher compression ratio can be achieved at a cost of higher PRD, when using a more complex JPEG2000 compression algorithm. Furthermore, fixed-point DCT models produce similar quality results at a CR 4:1. These results proves that fast DCT based data compression can be practically implemented in low cost, low power portable devices in order to decrease wireless communication bandwidth.

6. Conclusions

In this paper, we investigate the use of fast Discrete Cosine Transform algorithms for EEG data compression. We compare performance of fast DCT algorithms against direct DCT implementation. Experimental results showed a quality decrease of about 1%, with a significant speed increase when using fast algorithms. A multiplierless BinDCT algorithm produced the best compression speed result, with an insignificant loss of reconstruction quality.

Reconstruction quality, expressed in percent of root-mean-square difference (PRD), ranges from 5% to 11% at compression ratio 4:1 depending on the data used. These quality results are in line with other authors, using more complex compression algorithms. These results show that our purposed fast method for compression is efficient as well.

We conclude that fast fixed-point DCT algorithms can be used for EEG data compression however, applications are limited to low power embedded systems or wireless transmission for portable EEG devices.

Acknowledgments. The work described in this paper has been carried out within the framework the Operational Programme for the Development of Human Resources 2007-2013 of Lithuania "Strengthening of capacities of researchers and scientists" project VP1-3.1-SMM-08-K-01-018 "Research and development of Internet technologies and their infrastructure for smart environments of things and services" (2012- 2015), funded by the European Social Fund (ESF).

References

1. Berger, H.: Über das Elektrenkephalogramm des Menschen (On the EEG in humans). Archive für Psychiatre und Nervenkrankheiten 87, 527 – 570 (1929)
2. Birvinskas, D., Jusas, V., Martisius, I., Damasevicius, R.: EEG dataset reduction and feature extraction using Discrete Cosine Transform. In: Proceedings of the 6th European Modelling Symposium on Mathematical Modeling and Computer Simulation. pp. 186–191. IEEE Computer Society (2012)
3. Birvinskas, D., Jusas, V., Martisius, I., Damasevicius, R.: Data compression of EEG signals for artificial neural network classification. Information Technology And Control 42(3), 238–241 (2013)
4. Cardenas-Barrera, J.L., Lorenzo-Ginori, J.V., Rodriguez-Valdivia, E.: A wavelet-packets based algorithm for EEG signal compression. Medical Informatics and the Internet in Medicine 1, 15 – 27 (2004)

5. Chen, W.H., Smith, C.H., Fralick, S.: A Fast Computational Algorithm for the Discrete Cosine Transform. IEEE Transactions on Communications 25, 1004 – 1009 (1977)
6. Daou, H., Labeau, F.: Pre-Processing of multi-channel EEG for improved compression performance using SPIHT. In: Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE. pp. 2232–2235 (Aug 2012)
7. Dauwels, J., Srinivasan, K., Reddy, M., Cichocki, A.: Near-Lossless Multichannel EEG Compression Based on Matrix and Tensor Decompositions. Biomedical and Health Informatics, IEEE Journal of 17(3), 708–714 (May 2013)
8. Fira, M., Goras, L.: Biomedical Signal Compression Based on Basis Pursuit. In: Proceedings of the 2009 International Conference on Hybrid Information Technology. pp. 541–545. ICHIT '09, ACM, New York, NY, USA (2009)
9. Higgins, G., Faul, S., McEvoy, R., McGinley, B., Glavin, M., Marnane, W., Jones, E.: EEG compression using JPEG2000: how much loss is too much? In: 32nd Annual International Conference of the IEEE EMBS. pp. 614 – 617 (2010)
10. Higgins, G., McGinley, B., Faul, S., McEvoy, R., Glavin, M., Marnane, W., Jones, E.: The Effects of Lossy Compression on Diagnostically Relevant Seizure Information in EEG Signals. Biomedical and Health Informatics, IEEE Journal of 17(1), 121–127 (Jan 2013)
11. Kurbalija, V., Radovanović, M., Geler, Z., Ivanović, M.: The influence of global constraints on similarity measures for time-series databases. Knowledge-Based Systems 56(0), 49–67 (2014)
12. Liang, J., Tran, T.D.: Fast Multiplierless Approximations of the DCT With the Lifting Scheme. IEEE Transactions On Signal Processing 49, 3032 – 3044 (2001)
13. Loeffler, C., Lightenberg, A., Moschytz, G.S.: Practical fast 1-D DCT algorithms with 11 Multiplications. In: Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP 89). pp. 988 – 991 (1989)
14. Madan, T., Agarwal, R., Swamy, M.: Compression of long-term EEG using power spectral density. In: Engineering in Medicine and Biology Society, 2004. IEMBS '04. 26th Annual International Conference of the IEEE. vol. 1, pp. 180–183 (Sept 2004)
15. Martisius, I., Birvinskas, D., Jusas, V., Tamosevicius, Z.: A 2-D DCT hardware codec based on Loeffler algorithm. Electronics and Electrical Engineering 7, 47 – 50 (2011)
16. Memon, N., Kong, X., Cinkler, J.: Context-based lossless and near-lossless compression of EEG signals. Information Technology in Biomedicine 3(3), 231–238 (1999)
17. Mostow, J., Chang, K.M., Nelson, J.: Toward Exploiting EEG Input in a Reading Tutor. In: Proceedings of the 15th International Conference on Artificial Intelligence in Education. pp. 230–237 (2011)
18. Poh, K.K., Marziliano, P.: Compressive Sampling of EEG Signals with Finite Rate of Innovation. EURASIP Journal on Advances in Signal Processing 2010(1) (2010)
19. Salomon, D.: Data Compression. The Complete Reference. Springer, 3rd edn. (2004)
20. Sriraam, N.: Quality-on-Demand Compression of EEG Signals for Telemedicine Applications Using Neural Network Predictors. International Journal of Telemedicine and Applications 2011, 13 (2011)
21. Sriraam, N., Eswaran, C.: An adaptive error modeling scheme for the lossless compression of EEG signals. Information Technology in Biomedicine 12(5), 587–594 (2008)
22. Vidal, J.J.: Toward Direct Brain - Computer Communication. Annual Review of Biophysics and Bioengineering 2, 157–180 (1973)
23. Wongsawat, Y., Oraintara, S., Tanaka, T., Rao, K.: Lossless multi-channel EEG compression. In: Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on. pp. 4 pp.–1614 (May 2006)
24. Ylostalo, J.: Data compression methods for EEG. Technology and Health Care 7(4), 285 – 300 (1999)

Darius Birvinskas graduated at the Faculty of Informatics, Kaunas University of Technology (KTU) in Kaunas, Lithuania in 2009, where he received a B.Sc. degree in Informatics Engineering and his M.Sc. in 2011. Now, he is studying Ph. D. at the same University. He is also a lecture at Department of Computer Science, KTU where he lectures digital logic and computer architecture. He is author and co-author of several papers. His research interests include brain-computer interface and VLSI circuits design.

Vacius Jusas graduated from Kaunas Polytechnic Institute (Lithuania) in 1982. He received the D.Sc. degree from Kaunas Polytechnic Institute in 1988. Since 2006, he is professor at Department of Software Engineering, Kaunas University of Technology, Lithuania. He is author and co-author of more than 70 papers. He is Editor of journal Information Technology and Control. His research interests include testing of VLSI circuits, brain computer interfaces, adaptive signal processing.

Ignas Martisius graduated at the Faculty of Informatics, Kaunas University of Technology (KTU), Lithuania in 2009, where he received a B.Sc. degree in Informatics Engineering. He finished his M.Sc. studies in 2011. Currently, he is studying Ph. D. at the same University. Since 2014, he is a lecture at Department of Computer Science, KTU where he lectures digital logic and robot programming. He is author and co-author of several papers. His research interests include brain-computer interface and machine learning.

Robertas Damasevicius graduated at the Faculty of Informatics, Kaunas University of Technology (KTU) in Kaunas, Lithuania in 1999, where he received a B.Sc. degree in Informatics. He finished his M.Sc. studies in 2001 (cum laude), and he defended his Ph.D. thesis at the same University in 2005. Currently, he is a Professor at Software Engineering Department, KTU and lectures robot programming and software maintenance courses. His research interests include brain-computer interface, bioinformatics, data mining and machine learning. He is the author or co-author of over 100 papers as well as a monograph published by Springer.

Received: January 1, 2014; Accepted: August 5, 2014.