Parallel Implementation on GPU for EEG Artifact Rejection by Combining FastICA and TQWT

Afef Abidi
Technology and Medical Imaging Laboratory
Faculty of Medicine Monastir University of Monastir,
Monastir 5019, Tunisia
afefabidisa@gmail.com

Ibtihel Nouira
Technology and Medical Imaging Laboratory
Faculty of Medicine Monastir University of Monastir,
Monastir 5019, Tunisia
ibtihelnouira@gmail.com

Mohamed Hédi Bedoui
Technology and Medical Imaging Laboratory
Faculty of Medicine Monastir University of Monastir,
Monastir 5019, Tunisia
medhedi.bedoui@fmm.rnu.tn

***Abstract*—In this work, a new method for removal of ocular and muscular artifacts from Multi-channel electroencephalogram (EEG) is presented in order to obtain a 3D filtered cerebral mapping images. First, a FastICA algorithm of Independent Component Analysis (ICA) is applied in combination with the tunable Q-factor wavelet transform (TQWT), FastICA-TQWT. Then, to show the robustness of this method, a comparison was made between the proposed FastICA-TQWT method and the classical one FastICA-DWT based on three criteria: Mean Squared Error (MSE), correlation coefficient and Signal Noise Ratio (SNR). The results showed that the FastICA–TQWT method gave the highest Signal Noise Ratio and correlation coefficient and the minimum Mean Squared Error. However, the FastICA–TQWT algorithm requires an extremely high computing power. Therefore, the second contribution of this paper is to provide an EEG signal treatment by implementing the hybrid FastICA–TQWT algorithm using a new computing technology designed for a high-performance computing, called Graphical Processing Units (GPUs) using the Compute Unified Device Architecture (CUDA) technology. The performance of the parallel approach running along the GPU was compared to a CPU implementation.**

Keywords—Electroencephalogram; FastICA-TQWT; Mean Squared Error; correlation coefficient; Signal Noise ratio; Graphical Processing Units.

I. INTRODUCTION

Electroencephalogram (EEG) recordings are a kind of bioelectric signal which is collected by the electrodes on the surface of the scalp (due to the effect of millions of neurons and synapses) and reflects the internal state of the brain. This pattern of EEG is very helpful for the physician/doctor for the purpose of diagnosis such as tumor, finding of epileptic seizure and other medical disorders [1]. The EEG signal acquisition process is very susceptible to muscle, ocular and other noise interference [2]. These artifacts are required to be separated from the EEG signals to obtain the accurate results.

Recently, based on electroencephalogram (EEG) signals, there has been a growing interest in the development of reliable methods to reduce the artifacts while losing the minimum of useful information incorporated in the EEG [3, 4, 5, 6, 7]. For example, the authors in [4] presented the empirical decomposition approach (EMD). Furthermore, they reviewed recent methods of artifact removal based on the EMD and showed that the obtained results are very close to those found by classical methods. For automatic learning approaches, the authors in [5] have proposed a fuzzy cluster approach supporting the decision-making process by imitating the human learning process and estimated the similarity of the data leading to a non-iterative process. In addition, the work in [6] introduced an artificial neural network as a filter to suppress ocular artifacts. Other works used the Independent Component Analysis (ICA) algorithm versions (FastICA, Robust ICA, SOBI, extended Infomax and JADE) for the automatic artifact removal in EEG before processing signals to detect seizures [7, 8, 9, 10, 11]. In fact, the Principal Component Analysis (PCA) algorithm has demonstrated its effectiveness in removing imaging artifacts and BCG[9]. The wavelet transform, followed by ICA, has also been used for removing certain artifacts [10]. Furthermore, other works have used FastICA, which is an efficient, and popular algorithm applied to solve blind source separation (BSS) problems. This technique is widely used to identify artefacts and interferences from their mixtures such as EEG signal [11]. More recently, online recursive ICA algorithms have proven to be fast enough for real-time EEG source separation. However, none of them is automatic to identify artifacts among sources [12].

However, the implementation of these EEG signal filtering methods is far from being easy to implement on machines given the technological limitations and algorithmic complexity of different filtering tasks. As a solution, the graphic processing units (GPU), that exceeds the traditional computer systems, have reached sufficient maturity levels to enable robust operations [13]. The modern graphic processing units are not only powerful graphic engines but also parallel programmable processors that exceeds appreciably its equivalent processor in terms of bandwidth memory and computation acceleration [14]. In fact, many recent works have implemented different EEG processing applications on the GPU graphic processor, particularly the EEG inverse problem [15] and the brain imaging applications [16]. In addition, some works have been interested to implement the filtering

978-1-5386-9120-5/18/$31.00 ©2018 IEEE

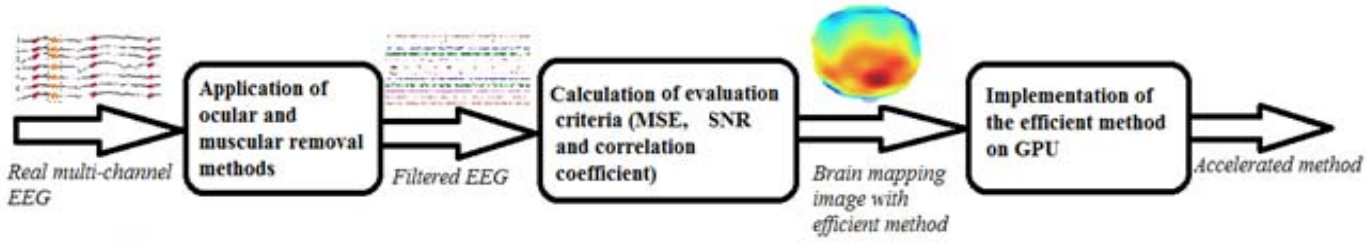

Fig.1. Block diagram of the processing system for EEG artifact rejection.

algorithms on GPUs in order to improve the execution time. Unfortunately, these works have been interested in the implementation of older versions of filtering methods, especially the Independent Component Analysis (ICA) methods [17, 18].On the one hand, the authors in [17] have replaced the basic linear algebra functions of Infomax-ICA algorithm to the standard CUBLAS routines provided by GPU manufacturers. On the other hand, they have implemented a new GPU-based solution, named CUDAICA. This solution is based on hybrid ad-hoc method for GPU optimizations. The results have showed an increase of 25x performance by using CUDAICA, over the standard ATLAS implementation, and 4.5x performance rise with regard to the MKL (Math Kernel Library) implementation.

However, now, the implementation of recent versions of EEG signal pretreatment methods on the GPU is a novelty. In this paper, the objective is to realize a novel approach for artifact removal from EEG data by combining FastICA and tunable Q-factor wavelet transform (TQWT). In addition, the second goal is to implement this method on a Graphical Processing Units GPU in order to meet firstly to specific clinical objectives and secondly to satisfy the requirement of real time processing. The present paper sheds some light on the acceleration of the ocular and muscular removal method to generate, in real time, a brain-mapping image showing the variation of useful cerebral activity.

This paper is organized as follows: in section 2 the basic concepts of artifact removal approach are presented. In this section, the different used methods, the evaluation criteria and the implementation process on GPU are detailed. In Section 3, the used EEG dataset is described. Finally, the experimental results and concluding remarks are presented in the last section.

II. METHODS

The pretreatment process implemented in the present work consists of three main blocks: application of ocular and muscular artifact removal methods, calculation of evaluation criteria (MSE, SNR, correlation coefficient) and finally the implementation of the efficient method on GPU (fig.1).

A. Ocular and muscular removal methods

In this section, two methods will be set in competition to eliminate the ocular and muscular artifacts. The former is the classical FastICA-DWT method. The latter is the proposed hybrid FastICA-TQWT method.

1) Fast Fixed Point Algorithm for ICA: The FastICA algorithm is one of the highly efficient method for performing the estimation of ICA that is based on demixing a statistically independent mixed signal [19]. In fact, the Independent Component Analysis ICA is a technique for separating a signal multivariate into subcomponents. The objective of the ICA method is to find a linear transformation of the given data. The following ICA model is expressed by (1) [20]:

$$X = AS + v \tag{1}$$

where:

- $X = (x_1, x_2, ..., x_m)$ is the random vector containing the mixed signal;

- $S = (s_1, s_2, ..., s_m)$ is the vector of independent source signal ;

-A is the unknown mixing matrix;

-v is the additive noise.

In order to reconstruct the exact vector S, the v value is assumed to zero. The ICA model is given by the following formula [20]:

$$X = AS \tag{2}$$

The goal of ICA is to estimate the linear transformation W of X with:

$$W = A^{-1} \tag{3}$$

So:

$$S = WX \tag{4}$$

To maximize the non-gaussianity (independence) of the sources, an un-mixing matrix W was searched. In FastICA algorithm, the non-gaussianity is measured with neg-entropy (J) approximation. The J approximation takes the following form:

$$J(y) = [E\{G(y)\} - E\{G(v)\}] \tag{5}$$

The un-mixing matrix of each component is given by the following algorithm [21]:

Step1: Randomize an initial row vector

Step 2: Apply Newton-Raphson method [22] :

$$W_i = E\left\{\hat{x}\, g'(W_i^T \hat{x})\right\} - E\left\{g''(W_i^T \hat{x})\right\} W_i \qquad (6)$$

with:

- g is the nonlinear function, where:

$$g = \frac{1}{\alpha_1}\log\cosh(\alpha_1 y) \qquad (7)$$

-g' and g'' are respectively the first and second derivation of g.

Step3: Normalize total matrix calculate after the above iterations:

$$W_i = \frac{W_i - mean}{std}.deviation \qquad (8)$$

Step 4: Apply the de-correlation formula:

$$W_i = W_i - \sum W_i^T W_j W_j \qquad (9)$$

Step 5: Repeat step 3

Step 6: If $W_i^T * W_{i-1}^T$ is not close enough to 1, calculate W_{i+1} and goes back to step 2.

2) Wavelet:

a) *Discrete Wavelet Transform:* The principle of the DWT is based on the decomposition and the reconstruction of the signal. At the decomposition phase, the DWT is changed into a tree of a filter bank using the multi-level decomposition by a high pass filter g(n) and a low pass filter h(n) (Fig. 2) [23]. At each level, the signal is decomposed into two elements: approximation (Aj) and detail (Dj). The former represents the general signal form or low frequency elements. The latter represents short events or high frequency elements.

The approximation and detail coefficients are respectively defined by (10) and (11).

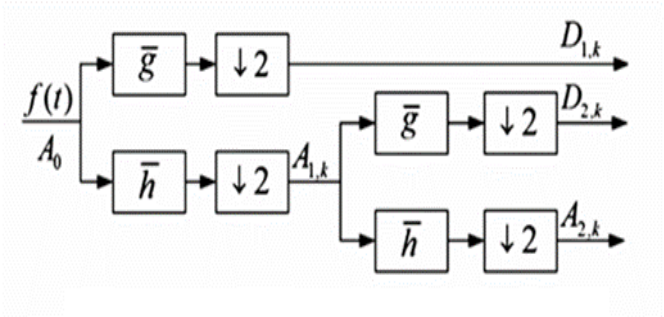

Fig.2.Mallat algorithm signal decomposition process diagram [24].

$$A_{j+1}(k) = \sum_{n=-\infty}^{\infty} h(n-2k)A_j \qquad (10)$$

$$D_{j+1}(k) = \sum_{n=-\infty}^{\infty} g(n-2k)A_j \qquad (11)$$

where j is the decomposition level.

b) *Tunable Q-factor wavelet transform:* The tunable Q-factor wavelet transform (TQWT) technique is utilized for signal decomposition in order to analyze the biomedical signals [25, 26]. This technique is powerful with three input parameters for analysis and the processing of oscillatory signals. These parameters, which are of quality-factor Q, redundancy r and a number of level j, are adjustable. The Q-factor is related to the bandwidth of a signal as [27]:

$$Q = \frac{f_0}{bw} \qquad (12)$$

The TQWT method can be implemented using two-channel filter banks iteratively. The expression of frequency response of a low-pass filter can be given as [27]:

$$H_0^{(j)}(w) = \begin{cases} \prod_{n=0}^{j-1} H_0\left(\frac{w}{a^n}\right), & |w| \le \alpha^j \pi \\ 0, & \alpha^j \pi \prec |w| \le \pi \end{cases} \qquad (13)$$

The expression of the frequency response of a high-pass filter can be given as [27]:

$$H_1^{(j)}(w) = \begin{cases} if\ (1-\beta)\alpha^{j-1}\pi \le |w| \le \alpha^{j-1}\pi \\ H_1\left(\frac{w}{a^{j-1}}\right)\prod_{n=0}^{j-2} H_0\left(\frac{w}{a^n}\right), \\ if\ -\pi \prec |w| \le \pi \\ 0, \end{cases} \qquad (14)$$

The fig. 3 shows the general system for j[th] level TQWT decomposition of an input signal s[n] to generate the low-pass sub-band signal cj [n] and the high-pass sub-band signal dj[n], respectively with sampling frequencies αf_s and βf_s :

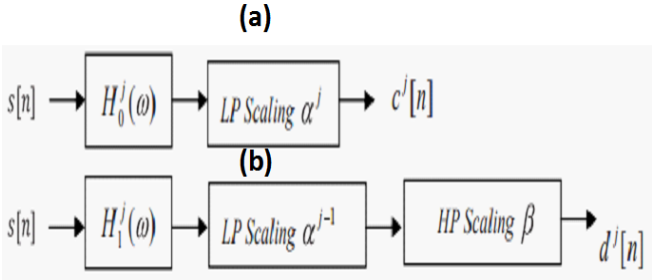

Fig.3. the [jth] level based decomposition to: (a) the low pass subband signal $c^j(n)$, (b) the high pass subband signal $d^j(n)$[28].

where:

- f_s the sampling frequency of signal x;

-α the low-pass scaling to generate $c^0[n]$;

-β the high pass scaling to generate $d^1[n]$.

The α and β factors respect the following conditions: $0<\alpha<1$, $0<\beta<1$ and $\alpha+\beta> 1$, in order to ensure a perfect reconstruction and to avoid redundancy. In fact, these two parameters are related to the scaling factors r and Q by respecting the following formulas [27]:

$$r = \frac{\beta}{1-\alpha} \tag{15}$$

$$Q = \frac{2-\beta}{\beta} \tag{16}$$

In this work, the best result has been reached with Q=3, r=3 and j=21.

B. Evaluation criteria

In this paper, the evaluation of the different ocular and muscular removal methods was achieved by measuring three evaluation criteria: Mean Squared Error (MSE), correlation coefficient and Signal Noise ratio (SNR).

Let x be the original signal, x′ the denoised signal and N the number of samples of EEG signal.

1) *Mean Square Error (MSE):* It is estimated between the denoised EEG signal and the original EEG signal. MSE is calculated using the following formula:

$$MSE = \frac{\sum_{i=1}^{N}(x(i)-x'(i))^2}{N} \tag{17}$$

2) *Signal to Noise Ratio (SNR):* SNR is defined as the ratio of power between the signal and the unwanted noise. SNR is defined as (18) [29]:

$$SNR = 10\log\left[\frac{\sum_{i=1}^{N}x(i)^2}{\sum_{i=1}^{N}\sum_{i=1}^{N}(x(i)-x'(i))^2}\right] \tag{18}$$

3) *The correlation coefficient:* The correlation coefficient measures the similarity between the denoised EEG signal and the original EEG signal and it is given by 19 [30]:

$$r = \frac{\sum_{i=1}^{N}(x(i)-\overline{x(i)})(x'(i)-\overline{x'(i)})}{\sqrt{\sum_{i=1}^{N}(x(i)-\overline{x(i)})^2\sum_{i=1}^{N}(x'(i)-\overline{x'(i)})^2}} \tag{19}$$

The noisy signal and the original signal are considered similar if the correlation coefficient is close to 1.

C. Implementation on GPU

When it comes to high-performance programming, architecture and parallel programming, they are the first things that come to mind. Mainly different types of parallelism and different parallel architectures are distinguished [31, 32]. This paper is orienting this research work towards the use of graphic processors, which are complex parallel devices to be programmed but capable of delivering a high performance.

Artifact removal methods have been set to enhance their quality by reducing the noise of artifacts. However, the applications of EEG artifact removal require a huge amount of computing power because of the high volume of EEG data and the intensive computing of the pretreatment methods. These properties have recently inspired the use of the massive parallel processors included in the graphical units (GPU) [33].

Firstly, the codes running on the GPU process are much faster than their serial counter parts. Secondly, they are less expensive to run than on standard parallel clusters. The availability of the state of art of the GPU that cards can play a promising role in accelerating the calculations of artifact removal methods.

The implementation process consists of creating a clean version of an EEG artifact rejection algorithm. GPU data processing is performed by a simple instruction on multiple data. The same operations are performed on a set of data in parallel.

The communication between the host (CPU) and the device (GPU) when running a CUDA program requires seven steps, as described explicitly below:

1-Allocate the parameters for the kernel code

2-Copy the parameters to the GPU memory

3-Call the kernel function.

4- Run the algorithm on the CUDA architecture

5- Return of the GPU / Kernel mode to the CPU / host mode

6-Copy the results.

7- The memory allocated in the GPU's global memory must be released in order to have not GPU memory leaks.

In this work, three GPU implementation methods have been used so as to benefit from the GPU Matlab.

The first technique is to utilize the MATLAB's parallel computing toolbox (PCT) that is based on the "arrayfun" function with gpuArray-type inputs. In fact, the "gpuArray"

command declares an array in the GPU memory. In this case, all matrix assignments will be executed directly on the GPU. The fastest option of matrix calculus is to utilize 'arrayfun' function. This latter implements a function on each value of an input matrix in order to store the results on an output matrix having the same size as the input one. To transfer the calculated results from the GPU memory to the MATLAB, the function "gather" is used.

The second method is the use of CUDA without .mex file. This method is based on creating a CUDA file (.cu), which contains the parallel CUDA Kernel object. Consequently, MATLAB compiles this file to create a .PTX (Parallel Thread Execution) file. In the next step, MATLAB uses CUDA file to search for a function prototype for the CUDA Kernel, which is defined in .PTX file by using the "parallel.gpu. CUDA Kernel" instruction. The evaluation of the Kernel on the GPU is done by the feval command.

The third method consists of using the CUDA with .mex file. After creating a .mex file, an analysis of the inputs/outputs and necessary memories allowance are performed. After that, a sequential calculation is performed in the CPU, however the parallel computation (loop for) is executed in the GPU. Finally, the results will be transfered to the matlab and a memory deallocation is realized.

In fact, the cost of creating and dispatching the thread on GPU is negligible because it is implemented by the hardware. But, Due to the bandwidth of PCIe (Peripheral Component Interconnect express) bus, the transmission of data between GPU and CPU is very time consuming. Therefore, it is necessary to increase as possible the transfer of data. In addition, the CPU is responsible for the complex logic control and the GPU is responsible for the intensive computation. So, it is possible to increase the overall speed significantly by using the GPU and the CPU simultaneously.There are several types of memory on GPU, such as register, shared memory, constant memory, texture memory, global memory… There are two ways in CUDA programming. The first one is the parallelism between different thread blocks where the global memory was used. The second one is the parallelism between threads in the same block by using the shared memory. In this work, the most crucial factor of implementation is the shared memory because this memory reduces memory access latency and enhances greatly the performance of the GPU algorithm. The proposed method is a good way to take the advantage of GPU, especially their hardware, and abundant CUDA for accelerating the code without losing the benefits of MATLAB.

III. Materials

The proposed artifact removal methods were developed using two different EEG datasets. The first dataset was characterized by the presence of muscular and ocular artefacts. This dataset was collected from 3 normal volunteers (2 males, 1 female) who were aged between 20 and 30 years. These EEG signals were recorded using 10-20 electrodes (19 channels), established by the standard international system with a 256 Hz sampling frequency. Each recording was 60s long.

The second dataset is of 6 epileptic subjects (2 males, 4 females). This dataset was characterized by the presence of generalized and localized waves and were recorded by 16 electrodes using 10/20 international system with a sampling frequency of 175 Hz. Each recording of the second dataset was of 10s long [34].

IV. Results

A. Performance according to evaluation criteria

Fig. 4 presents an example of rejecting eye blink and muscular artifacts. In fig.4 (a), the eye-blinking and muscular artifacts appears in EEG as big pulses. It is well localized in time and has the strongest impact on EEG signals. To remove these noises both FastICA-TQWT and FastICA-DWT filters were applied respectively in fig.4 (b) and fig.4 (c). According to expert knowledge and EEG records displayed in fig.4 (b) and fig.4 (c), FastICA-TQWT can be the best choice for noise removal.

To quantify the quality of recovering of the two cerebral datasets, the average of MSE, RSB and correlation coefficient between the real EEG in the different channels and the signals obtained after FastICA–DWT and FastICA-TQWT processing were calculated.

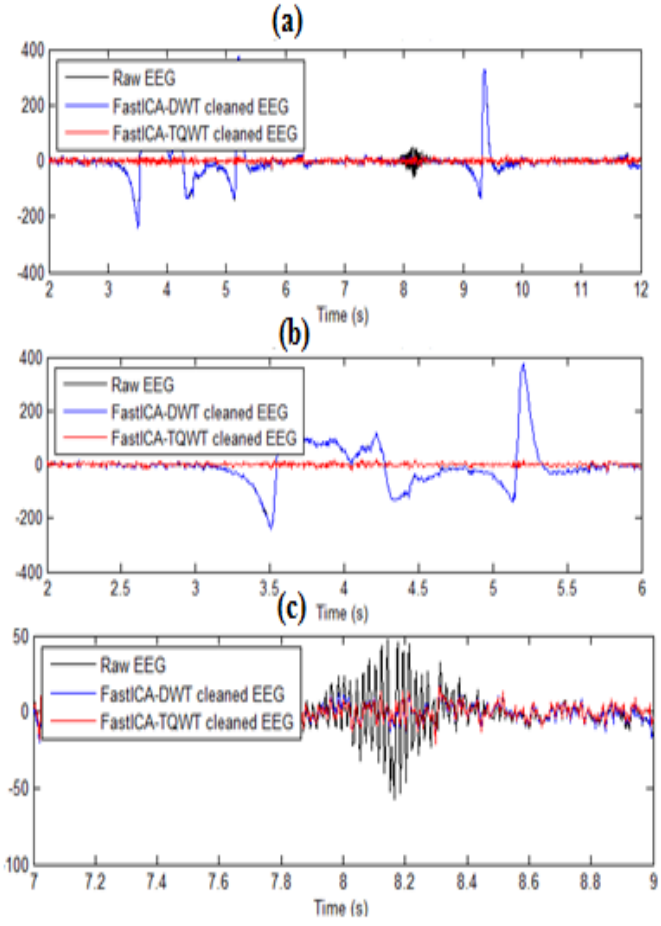

Fig.4. Filtered Fp1 EEG signal: (a) all artefacts, (b) muscular artefacts, (c) ocular artefacts.

TABLE I. COMPARISON OF MSE, SNR AND CORRELATION COEFFICIENT OF DIFFERENT METHODS FOR THE FIRST AND SECOND DATASETS.

Subject	MSE		SNR		Coefficient correlation	
	M.1	*M.2*	*M.1*	*M.2*	*M.1*	*M.2*
1	0.478	0.142	10.001	14.030	0.973	0.981
2	0.581	0.203	12.332	19.211	0.985	0.997
3	0.493	0.159	15.941	19.561	0.950	0.982
A0004	0.191	0.077	3.701	5.302	0.997	0.999
A0007	0.166	0.090	4.499	6.332	0.982	0.993
A0010	0.156	0.069	2.55	4.001	0.99	0.995
A0011	0.188	0.101	2.441	3.667	0.984	0.994
A0015	0.21	0.113	3.931	6.002	0.988	0.996
A0016	0.199	0.098	3.772	5.438	0.978	0.989

The results from TABLE. I show that FastICA-TQWT (M.2) is better than the FastICA-DWT method (M.1) in terms of the accuracy of filtering by giving the highest values of SNR, correlation coefficients and the lowest value of MSE in both cases of healthy and epileptic subjects. This reveals the great performance of the FastICA-TQWT method.

Fig.5 shows the mapping at the time of the crisis (t = 5s). Fig.5 (a) is signal mapping before filtering and fig.5 (b), (c) (d) are signal mapping images after filtering using respectively the FastICA/ FastICA-DWT/ FastICA–TQWT methods. These results confess the diagnosis of the expert who confirmed that the crisis is focused on the right bilateral zone.

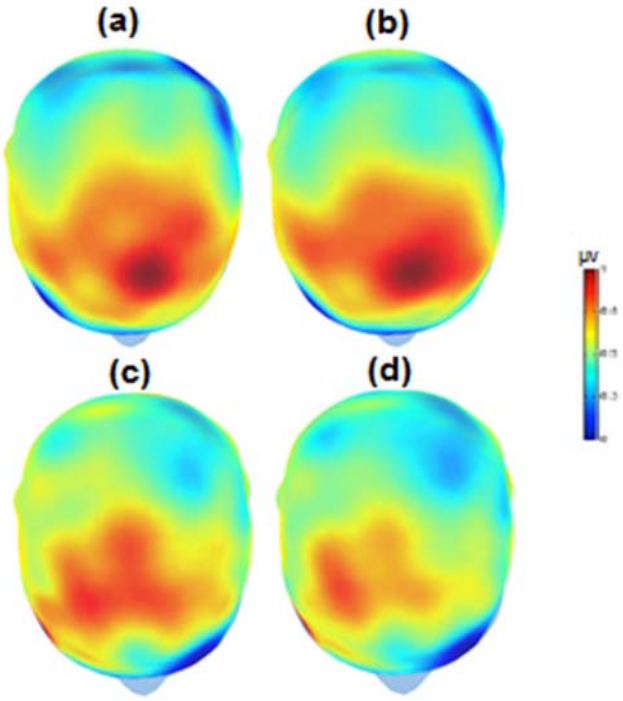

Fig. 5. Cartography at the time of seizure (t=5s): (a) real EEG signal; EEG after filtering using: (b) FastICA (c) FastICA-DWT (d) FastICA-TQWT.

Based on these results, in the rest of the work, the FastICA–TQWT method is adopted to ocular and muscular artifacts removal. This method needs to be speeded up by implementing it on the GPU to ensure fast execution of the filtration process. The next sub-section gives the results obtained from different techniques of the implementation of this hybrid method on GPU.

B. Implementation of FASTICA-TQWT on GPU

All calculations reported in this paper were performed on an Intel(R) Core(TM) i3-4005u CPU 1.70 GHz, and Nvidia Geforce 820M. It is a Fermi GPU, with 96 CUDA cores, 2GB DDR3; 1800 MHz device memory, single precision float point performance where the Matlab version is R2012a, the Visual studio version is 2010 prof and SDK is CUDA 5.0.

Since the FastICA-TQWT method represents the best filtering results, a real time implementation on GPU of this method was realized. The execution time of FastICA-TQWT method available in CPU was compared with the processing time on GPU. Three types of implementation on GPU were used. The first one is the use of GPU-CUDA without .mex file. The second is the use of GPU with MATLAB through Parallel Computing Toolbox. Finally, the third method is to accelerate the code using mex programming and GPU CUDA.

Based on the result in fig.6, the time cost by CPU is significantly higher than the GPU. In fact, when MATLAB was used with c-mex and CUDA, the optimization rate reaches 49.74%. While, with the parallel computing toolbox of MATLAB, the optimization rate is only 35.04%. However, the use of CUDA without .mex file method gives the wrong results by getting an execution time optimization of 9.34% only. In current a situation, GPU utilization through c-mex proves a better approach to accelerate the FastICA-TQWT implementation.

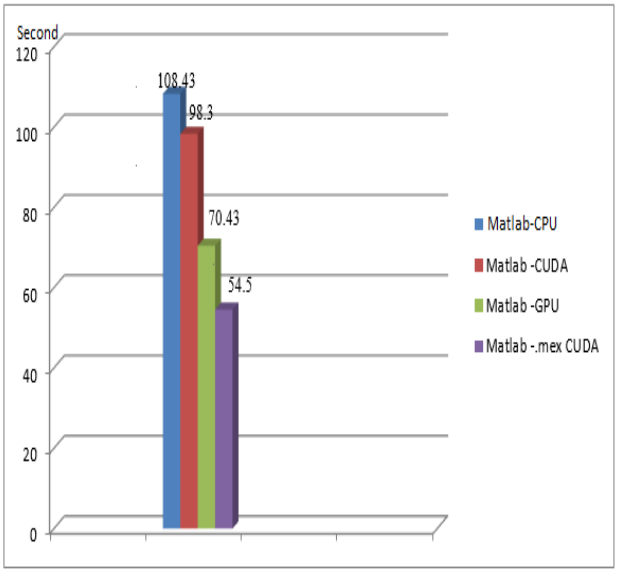

Fig. 6. Comparison of the FastICA-TQWT execution time between an execution in the CPU and the three GPU acceleration methods.

V. CONCLUSION

EEG signal may usually be contaminated by several sources of artifacts like ocular and muscular artifacts. These artifacts must be corrected or separated from clean EEG signal before further analysis because they make the analysis difficult. In this work, an automatic artifact removal hybrid system based FastICA combined with TQWT was used, named FastICA-TQWT. To evaluate this algorithm, a comparison between the proposed FastICA-TQWT and the FastICA-DWT method was evaluated by using three evaluation criteria (MSE, SNR and correlation coefficient). The results indicated that the FastICA-TQWT filtering method has a better performance than the FastICA-DWT. This technique will lead to better results while interfacing it as a pre-processing step during the biometric recognition. FastICA-TQWT could also be used for extraction of other features in biomedical signals. However, this algorithm was computationally intensive and time-consuming. To solve this problem, a GPU implementation of FastICA-TQWT method was taken using three techniques: CUDA and C++ with .mex file, CUDA without .mex file and MATLAB parallel computing toolbox. The results showed that the GPU implementation offers an optimization rate of 49.66% in the case of MATLAB with c-mex and CUDA implementation. These encouraging results allow us to plan further work to improve the execution time rate.

REFERENCES

[1] N. Ahammad, T. Fathima, and P. Joseph, “Detection of epileptic seizure event and onset using EEG,” BioMed Res Int 2014 (2014), 450573. Available at: http://dx.doi.org/10.1155/2014/450573.

[2] M. Teplan, "Fundamentals of EEG measurement," Measurement science review, vol. 2, pp. 1-11, 2002.

[3] N. Tibdewal Manish , R. Fate Rohan , M. Manjunatha , A. Kumar Ray, M. Malokar, “Classification of artifactual EEG signal and detection of multiple eye movement artifact zones using novel Time-amplitude algorithm,” Signal, Image and Video Processing, vol. 11(2), pp.333-340, 2017.

[4] D. Looney, V. Goverdovsky, P. Kidmose, and D. P. Mandic, "Subspace denoising of EEG artefacts via multivariate EMD," IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4688-4692, 2014.

[5] C. Bedoya, D. Estrada, S. Trujillo, N. Trujillo, D. Pineda, and J. D. López, "Automatic component rejection based on fuzzy clustering for noise reduction in electroencephalographic signals," Symposium of Image, Signal Processing, and Artificial Vision (STSIVA), 2013 XVIII, pp. 1-5, 2013.

[6] J. Mateo, A. M. Torres, and M. A. García, "Eye interference reduction in electroencephalogram recordings using a radial basic function," IET Signal Processing, vol. 7, pp. 565-576, 2013.

[7] M. De Vos, W. Deburchgraeve, P. Cherian, V. Matic, R. Swarte and P. Govaert, "Automated artifact removal as preprocessing refines neonatal seizure detection," Clinical Neurophysiology, vol. 122, pp. 2345-2354, 2011.

[8] N. Oosugia, K. Kitajoc, N. Hasegawaa, Y. Nagasakaa,K. Okanoyab and N. Fujii, "A new method for quantifying the performance of EEG blind source separation algorithms by referencing a simultaneously recorded ECoG signal,"Neural Networks, vol.93, pp.1-6, 2017.

[9] M. Negishi,M. Abildgaard, T. Nixon and R. T. Constable," Removal of time-varying gradient artifacts from EEG data acquired during continuous fMRI," Clinical neurophysiology, vol. 115, pp. 2181-2192, 2004.

[10] M.T. Akhtar, W. Mitsuhashi, and C.J. James, "Employing spatially constrained ICA and wavelet denoising, for automatic removal of artifacts from multichannel EEG data. Signal processing", vol. 92, pp. 401-416, 2012.

[11] T. Ahmad, H..N. Binti Alias and M. Ghanbari, "Separation of the EEG Signal using Improved FastICA Based on Kurtosis Contrast Function,"Australian Journal of Basic and Applied Sciences, vol.5, pp. 2152-2156, 2011.

[12] H.S. Hsu, T. R. Mullen, T.-P. Jung, and G. Cauwenberghs, "Real-time adaptive EEG source separation using online recursive independent component analysis" IEEE transactions on neural systems and rehabilitation engineering, vol. 24, pp. 309-319, 2016.

[13] D. Kirk, W. Hwu, “Programming Massively Parallel Processors: A Hands-onApproach. Book,” Morgan Kauffman Publishers Inc., 2010.

[14] M. Chouchene , F. E. Sayadi, M. Atri, R. Tourki, Integral image computation on GPU, the 10th International Multi-Conference on Systems, Signals & Devices (SSD), 2013, DOI: 10.1109/SSD.2013.6564007

[15] Juhasz Z, Kozmann G, “ A GPU-BASED SOFT REAL-TIME SYSTEM FOR SIMULTANEOUS EEG PROCESSING AND VISUALIZATION,”, Scalable Computing: Practice ans Experience, vol. 17, pp. 61-78, 2016.

[16] Z. Juhasz and G. Kozmann, “A GPU-based simultaneous real-time EEG processing and visualization system for brain imaging applications,” 38th International Conventionon Information and Communication Technology, Electronics and Microelectronics MIPRO 2015 - Proc., no. May, pp. 299–304, 2015.

[17] F. Raimondo, J.E. Kamienkowski, M. Sigman and D. Fernandez Slezak, “CUDAICA: GPU Optimization of Infomax-ICA,” EEG Analysis Computational Intelligence and Neuroscience 2012, vol. 2012, DOI 10.1155/2012/206972

[18] A. Foshati, F. Khunjush, A novel implementation of double precision and real valued ICA algorithm for bioinformatics applications on GPUs, Euro-Par'12 Proceedings of the 18th international conference on Parallel processing workshops, p. 285-294, 2012.

[19] Z. Koldovský, P. Tichavský, and E. Oja,” Efficient Variant of Algorithm FastICA for Independent Component Analysis Attaining the Cramér-Rao Lower Bound,” IEEE Transactions on Neural Networks, Vol. 17(5), pp1265-1277, 2006.

[20] A. Hyv, "Fast and robust fixed-point algorithms for independent component analysis," IEEE Transactions on Neural Networks, vol. 10, pp. 626-634, 1999.

[21] A. Palumbo, F. Amato, B. Calabrese, P. Vizza, An Embedded System for EEG Acquisition and Processing for Brain Computer Interface Applications, DOI 10.1007/978-3-642-15687-8_7,Wearable and Autonomous Biomedical Devices and Systems for Smart Environment: Issues and CharacterizationPublisher: Springer Verlag, Editors: S. C. Mukhopadhyay, A. Lay-Ekuakille, 2001, pp.137-154.

[22] S. Basiri; E. Ollila; V. Koivunen,"Alternative Derivation of FastICA With Novel Power Iteration Algorithm,"IEEE Signal Processing Letters, Vol. 24(9), pp. 1378 – 1382, 2017.

[23] A.R. Al Qawasmi and K. Daqrouq, "ECG Signal Enhancement Using Wavelet Transform," WSEAS Trans on Biology and Biomedecine, Vol. 7, pp. 62-72, Apr. 2010.

[24] S.G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," IEEE transactions on pattern analysis and machine intelligence, vol.11, pp. 674-693, 1989.

[25] S. Patidar, R.B. Pachori, U.R. Acharya, “Automated diagnosis of coronary artery disease using tunable-Q wavelet transform applied on heart rate signals,” Knowl.-Based Syst., vol. 82, pp. 1-10, 2015.

[26] A. Bhattacharyya, R.B. Pachori, A. Upadhyay, U.R. Acharya,”Tunable -Q wavelet transform based multiscale entropy measure for automated classification of epileptic EEG signals,” Applied Sciences, vol. 7(4), pp. 1-18, 2017.

[27] I.W. Selesnick,"Wavelet transform with tunable Q-factor,"IEEE Trans. Signal Process., vol.59, pp. 3560-3575, 2011.

[28] A.R. Hassan and M.I.H. Bhuiyan, "An automated method for sleep staging from EEG signals using normal inverse Gaussian parameters and adaptive boosting,"Neurocomputing, vol. 219, pp. 76-87, 2017.

[29] O.Väisänen, J. Malmivuo, “Improving the SNR of EEG generated by deep sources with weighted multielectrode leads,”Journal of Physiology-Paris,Vol. 103(6), pp. 306-314, 2009.

[30] S. Diwaker1, S. K. Gupta, N. Gupta,” Classification of EEG Signal using Correlation Coefficient among Channels as Features Extraction Method,” Indian Journal of Science and Technology, Vol 9(32), DOI: 10.17485/ijst/2016/v9i32/100742, 2016.

[31] M. McCool, S.Du Toit, Metaprogramming GPUs with Sh. AK Peters Ltd, 2004.

[32] J. Gray, The Past and Future of FPGA Soft Processors. FPGA CPU News, 2014

[33] M. Chouchene, F. E. Sayadi, H. Bahri, J. Dubois, J. Mitéran and M. Atri, “Optimized parallel implementation of face detection based on GPU component.Microprocessors and Microsystems”, Embedded Hardware Design, Vol. 39, pp. 393-404, 2015.

[34] http://www.unb.ca/cic/datasets/index.html