

On the implementation and improvement of automatic EEG spike detection algorithm

Andrius Vytautas Misiukas Misiūnas, Tadas Meškauskas,
Algimantas Juozapavičius

Faculty of Mathematics and Informatics, Vilnius University

Naugarduko 4, LT-03225 Vilnius

E-mail: andrius.misiukas@mif.stud.vu.lt, tadas.meskauskas@mif.vu.lt

E-mail: algimantas.juozapavicius@mif.vu.lt

Abstract. The algorithm of automatic EEG spike detection and its implementation is described in this article. The algorithm implemented is based on mathematical morphological filters which distinguishes background brain activity from EEG spikes. The implementation of the algorithm is system independent, it can be deployed on both personal computers and clusters with MPI parallel computing support.

Keywords: EEG, epilepsy, EEG spikes, computer vision.

Introduction

Signal analysis is widely used in modern medicine, electrocardiograms (ECGs) are probably the most widely known signal used in modern medicine. EEG is somewhat similar to ECG but it contains data of brain activity. Nowadays most of EEGs are acquired in 10–20 system contains signals of 21 electrodes which are placed on patient's head [3]. EEG might have some additional channels like ECG, electrooculograms, electromiograms and others.

Electroencephalography (EEG) is a brain activity monitoring technique widely used in diagnosis and monitoring of epilepsy and some other Central Nervous System (CNS) diseases. Manual examination of EEGs is very tedious and time consuming task for medical personnel, resulting in many automatic EEG analysis algorithms created, most of which are summarized by [1].

EEG spike is a short peak in patients brain activity, which consists peak and slow wave, which follows immediately after the peak. Typically EEG spike (together with the slow wave) is between 40 ms and 200 ms long [3]. In order to detect an EEG spike it must be detected independently in at least two EEG channels.

Automatic EEG spike detection existing today are based on a few mathematical principles. The main groups of these algorithms are template matching, parametric methods, mimetic analysis, power spectral analysis, wavelet analysis and artificial neural networks [1]. The algorithm implemented falls into it's own category which could be called reverse template matching since it filters out normal brain activity patterns and leaves EEG spikes only.

During this work algorithm by [2] was implemented and improved. Most of the algorithms described in [1] either does not generate satisfactory results either generate

it for some specific groups of patients. The algorithm by [2] does not rely on peak detection itself, but rather on removal of normal background brain activity from original EEG signal. Artefacts are filtered out of resulting signal and only EEG spikes are left.

Many of algorithms of automatic EEG analysis have their known sets of flaws and limitations. Some of the algorithms demonstrate a very high degree of accuracy but they are tested on data collected from group of patients with specific form of epilepsy. Algorithm selected here does not rely on EEG spike form itself rather it filters out normal background activity of brain from the EEG signal [2].

The main result of this work are some developments over the original algorithm described in [2] and the implementation of current algorithm version. The implementation works on both distributed computing platforms and personal computers as well. The algorithm is implemented using Python programming language, the implementation can work on any operating system, which has an implementation of Python interpreter.

1 EEG spike detection algorithm

1.1 Morphological operators

The algorithm described by [2] is based on series of mathematical morphological filters and operations. The most basic morphological operations used here are erosion and dilation.

The analysed EEG signal is denoted by $f(t)$. The structuring element is defined as $g(t)$ and reflection of structuring element $g^s(t) = g(-t)$. D is the domain of signal $f(t)$. Then erosion can be defined as:

$$(f \ominus g^s)(t) = \min_{\tau \in D} \{f(\tau) - g(- (t - \tau))\}. \tag{1}$$

Dilation:

$$(f \oplus g^s)(t) = \min_{\tau \in D} \{f(\tau) + g(- (t - \tau))\}. \tag{2}$$

Using these operators opening and closing operators can be defined. Opening:

$$(f \circ g)(t) = [(f \ominus g^s) \oplus g](t). \tag{3}$$

Closing:

$$(f \bullet g)(t) = [(f \oplus g^s) \ominus g](t). \tag{4}$$

Since EEG spikes can have both positive and negative amplitudes additional operators are defined. Close-opening:

$$CO(f(t)) = f(t) \bullet g_1(t) \circ g_2(t). \tag{5}$$

Open-closing:

$$CO(f(t)) = f(t) \circ g_1(t) \bullet g_2(t). \tag{6}$$

Open-closing increases average values of signal while close-opening decreases by same amount therefore an average of these values is used:

$$OCCO(f(t)) = (OC(f(t)) + CO(f(t)))/2. \tag{7}$$

1.2 Structuring elements

Structuring element is essential to distinguish between background brain activity and EEG spikes. Background EEG activity is best matched by parabola:

$$g_i(t) = a_i t^2 + b_i, \quad i = 1, 2. \quad (8)$$

Since brain rhythms can have different any frequencies between 0.5 Hz and 100 Hz and amplitudes, parameters a and b are different for each section of EEG signal. For that it is necessary to define parameter W , which is an array of EEG signal arc lengths. Then parameters a and b can be defined:

$$b_1 = b_2 = \text{median}(|f|), \quad (9)$$

$$a_{1,2} = \text{median}(|f|) / (k \cdot \text{median}(W)), \quad k = 0.5, 1.5. \quad (10)$$

During experiments it was found that optimal structuring element width is equal to $\text{median}(W)$.

1.3 Algorithm for spikes detection

This algorithm is further improvement of algorithm presented by [2]. This algorithm contains an evaluation length of EEG signal which is optimal for the analysis using the algorithm. Using experiments with real patients data it was established that optimal chunk of EEG data for one iteration of algorithm is between 4 and 6 seconds long. Main reason for that is background activity of patients brain is constantly changing and parameters a_i and b_i should be re-evaluated for each period of time. If parameters of structuring elements cannot be determined correctly, some spikes can get filtered out or artefacts of background brain activity can be still present in filtered signal.

Second argument for period chosen is that during longer periods of time, baseline of the EEG signal can change. This can happen because of sweat on patient's head which changes conductive properties of patient's skin which leads in changes of EEG signal baseline. Short period of time chosen lets eliminate these kind of artefacts more easily.

Morphological filters described in 1 section are applied directly to the chunks of EEG signal described in this chapter. In order to perform computations more efficiently the morphological filtering is done using:

$$F_{\text{filtered}}(t) = \text{abs}(F(t)) - \text{OCCO}(F(t)), \quad (11)$$

where $F_{\text{filtered}}(t)$ is the signal used in further computations, $F(t)$ is the original signal and $\text{OCCO}(F(t))$ is the filter described in 1 chapter. Any spikes over detection limit are considered as possible epileptiformic spikes.

During this study many values of EEG spike detection limit and optimal definition for available data was chosen. Since analysed chunks of data are quite short and does not contain large changes of average value of signal, we can use the definition similar to used by doctors where EEG spike is detected if it rises more than two times above the background activity of the brain:

$$L = 2.0 \text{median}(f). \quad (12)$$

This definition still generates some false detections, however in order to detect EEG spike it must be detected in at least two neighbouring EEG electrodes. This additional check eliminates almost all the false detections.

Overall computation time depends not only on the overall length of data, but on sizes of data chunks as well. Due to implementation details the choice of smaller times will lead to shorter overall computation time compared to large data chunks. It should be also noted that individual chunks data should contain enough data for determining of parameters of structuring element and correct operation of morphological filters.

2 Details of implementation

2.1 Programming language

Implementation of this algorithm was made using Python programming language. Python has full support of MPI protocol, unlike most of very high level object-orientated programming languages (e.g. Java). Second reason is that Python is a very high level language, but it has a lot of ready to use libraries which are partially written in C language what grants access to some C like speed functions from very high level APIs. Third reason is amount and quality of open source science orientated libraries. This grants smaller amount of legal issues when trying to deploy the algorithm for real world usage.

2.2 Paralellisation of the algorithm

The implementation of the algorithm was paralellised in order to benefit from advantages of distributed computing systems also known as supercomputers. The benefits from paralellisation can vary depending on length of EEG analysed. Analysis of 1 minute EEG record takes about 2 to 3 minutes of computing time on standard PC. Length on an EEG can vary widely ranging from 2 to 30 minutes, that means that usage of paralellisation saves significant amounts of time (up to 3 hours) for analysis of longer EEGs. Parallel version of the algorithm allows getting the answer within 15 minutes time margin regardless of length of EEG processed. Additionally it can be used through the web site without any software installation or setting up.

The paralellisation is implemented under MPI protocol, the analysis system has a single threaded launcher, which downloads the data, checks the integrity of the system and initial data file. If integrity tests are passed, sbatch MPI job is formed and submitted for analysis of the initial data file. During the launcher phase of the system, only one CPU is allocated.

The launcher takes into account length of initial data file and requests for an optimal number of processors for the work. After queuing required number of CPUs is allocated and the parallel section of the algorithm starts. After analysis is done all the data in the main thread where a report of analysis is constructed. Main thread eliminates all the repeating EEG spikes what could happen because of overlapping. The report then is sent back to the request server.

Because of method of paralellisation chosen, the system has a great EEG length – processor count scalability. With reasonable lengths of EEG, the speedup versus processor count is almost linear.

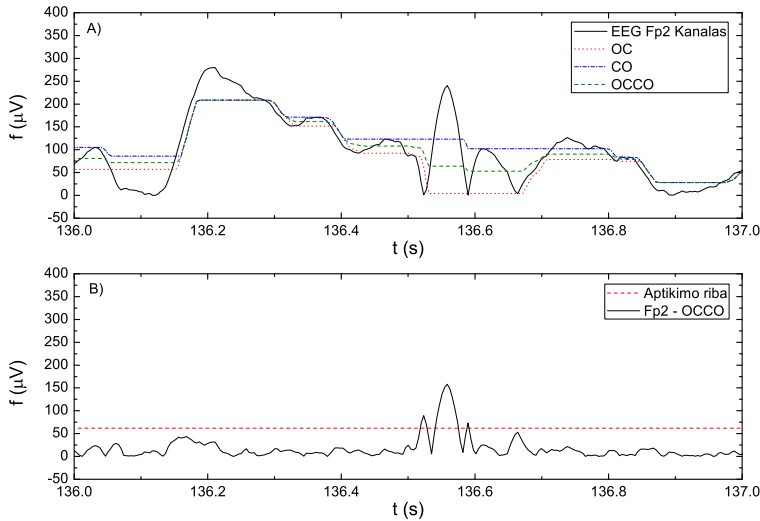


Fig. 1. Morphologic operations demonstrated. (A) Contains the original EEG channel, results of OC, CO and OCCO filters applied. (B) Contains difference between OCCO and filtered signal.

3 Experimental results

Performance and results of the algorithm implemented are discussed in this section. Fig. 1 demonstrates the basic working principles of the algorithm on sample of real data. It is visible that portion of signal between 136.1 and 136.5 contains some kind of data artefact. The algorithm successfully filters this data out, only insignificant residue is left in filtered data.

Some artefacts still can get over the detection limit. However another requirement helps them: in order to EEG spike be detected, it must be found in at least two neighbouring channels of the EEG. The processes rapid enough to cause false detections are usually confined to single EEG electrode. Too high detection limit worsens the results more than too low one, since all the peaks under the detection limit are not considered in further analysis, and falsely detected ones are considered but in most of the cases are rejected during the process of EEG analysis.

Fig. 1 contains three peaks, which are higher than the detection limit. The first of them is an artefact and is not detected in other EEG channels. The second of them is the real EEG spike, which is detected across multiple EEG channels. The third peak is the the ripple of the EEG spike and is regarded as single structure all together with the spike.

The implementation of algorithm was tested with real 203 real patient EEGs. All of EEGs were processed however the accuracy of the results somewhat depend on factors like signal to noise ratio of the EEG measurement, number of EEG artefacts and amplitude of EEG peaks. Highly artefacted data tends to give less EEG peaks than usual since the algorithm does not analyse parts of signal near the artefact. If measurements done with old and low quality equipment and other overwhelming artefacts are not included, doctors evaluated the reliability of the algorithm by about 90 percent, this would be 138 EEGs of our test data.

4 Results and conclusions

During this study further developments of original algorithm published in [2] were made, the current version of the algorithm is presented in this article.

The implementation of algorithm is platform independent, it can be deployed on distributed computing machines as well as personal computers as long as Python interpreter exists in deployment environment. The implementation scales almost linearly on increased number of CPUs.

The discussed algorithm focuses on detecting EEG spikes by filtering out normal brain background activity. This approach performs well on filtering out data artefacts.

The implemented algorithm is tested on real patient data provided by Childrens Hospital, Affiliate of Vilnius University Hospital Santariškių Klinikos. The system is now used by Lithuanian medical personnel through <http://nksp.eu/> web site.

Acknowledgments. We thank very much Rūta Samaitienė from Childrens Hospital, Affiliate of Vilnius University Hospital Santariškių Klinikos for providing the data for testing the algorithm and its implementation.

We also thank Nacionalinė Klinikinių Sprendimų Palaikymo Sistema (NKSPS) project for financing and granting the access to the system through <http://nksp.eu/> web site.

References

- [1] J.J. Halford. Computerized epileptiform transient detection in the scalp electroencephalogram: obstacles to progress and the example of computerized {ECG} interpretation. *Clin. Neurophysiol.*, **120**(11):1909–1915, 2009.
- [2] A. Juozapavičius, G. Bacevičius, D. Bugelskis and R. Samaitienė. EEG analysis – automatic spike detection. *Nonlinear Anal. Model. Control*, **16**(4):375–386, 2011.
- [3] S. Sanei and J.A. Chambers. *EEG Signal Processing*. John Willey & Sons, 2007.

REZIUMĖ

EEG pikų radimo algoritmo patobulinimai ir įgyvendinimas

A. V. Misiukas Misiūnas and T. Meškauskas and A. Juozapavičius

Diagnozuojant ir gydant epilepsiją bei kai kurias kitas centrinės nervų sistemos (CNS) ligas, reikia kompiuteriu apdoroti ir analizuoti pacientų elektroencefalogramas (EEG). Siekiant automatizuoti medikų darbą, buvo iširtas ir realizuotas EEG pikų atpažinimo algoritmas. Atsižvelgus į algoritmo veikimo rezultatus, įgyvendinti algoritmo patobulinimai bei optimizacijos. Darbe pristatomos algoritmo įgyvendinimo detalės ir veikimo pavyzdžiai.

Algoritmo pagrindą sudaro matematinės morfologijos filtrai, galutinio rezultato pagerinimui pritaikomi papildomi filtrai, pašalinantys duomenų artefaktus iš EEG. Taip pat algoritme apskaičiuojamas paciento mieguistumas Karolinska skalėje (tam, kad atskirti miego ir budrumo būsenoje darytas EEG). Be to, algoritmo vykdymo metu įvertinamos smegenų ritmų dažninės charakteristikos, galinčios turėti papildomos vertės nustatant tikslią diagnozę. Algoritmo pagrindu sukurtas kompiuterinės EEG analizės įrankis ne tik ženkliai palengvina bei pagreitina medikų darbą, tačiau ir leidžia apskaičiuoti parametrus, kurių įvertinimas rankiniu būtu sudėtingas. Programinė algoritmo realizacija (Python programavimo kalba) testuota tiek asmeniniuose kompiuteriuose, tiek ir paskirstytų skaičiavimų platformose, tame tarpe Vilniaus Universiteto Matematikos ir informatikos fakulteto Skaitmeninių tyrimų ir skaičiavimo centro superkompiuteriye. Įrankis naudojamas Vilniaus universiteto ligoninės Santariškių klinikų specialistų.

Raktiniai žodžiai: EEG, elektroencefalogramos, EEG pikai, epilepsija.