Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 46 (2015) 292 – 298

Procedia Computer Science

International Conference on Information and Communication Technologies (ICICT 2014)

Using Brain Computer Interface for Synthesized Speech Communication for the Physically Disabled

Sumit Soman*, B K Murthy

Centre for Development of Advanced Computing, Noida, India

Abstract

Brain Computer Interface (BCI) systems have been widely used to develop viable assistive technology for physically disabled persons. In this paper, we present the design and development of a BCI-based system for generation of synthesized speech, which works on eye-blinks detected from the Electroencephalogram (EEG) signals of the user. Such a system is particularly useful for patients suffering from locomotive disorders such as locked-in syndrome, who can use this interface to communicate with their caretakers. This system enables patients to communicate by selecting the desired options from a configured list by performing eye-blinks, which is then converted to synthesized speech by the computer system. The key advantages of our system are that it uses the portable and easy-to-wear Emotiv headset, is built on using an open-source application stack and also does not require training for individual users. The system has been tested on patients who have been able to use it conveniently to communicate with their caretakers in a medical facility.

Peer-review under responsibility of organizing committee of the International Conference on Information and Communication Technologies (ICICT 2014)

Keywords: Brain Computer Interface, Assistive Technology, Electroencephalogram, Emotiv EPOC Neuroheadset, Synthesized Speech

1. Introduction

Brain Computer Interface (BCI) enables the use of voluntary variations in brain activity to control devices. These systems typically use a combination of signal processing and machine learning techniques to identify intents from Electroencephalogram (EEG) signals, which are then translated to commands by an end-user application[1]. Such systems have recently gained popularity in being used as assistive technology for patients suffering from locomotive disorders such as Amyotrophic Lateral Sclerosis (ALS), locked-in syndrome and other neurological disorders[2]. BCI systems are useful to such patients, as they provide a medium for communication using actions discernible from the subject's EEG signals.

* Corresponding author. Tel.: +91-120-3063311 ; fax: +91-120-3063317.
E-mail address: sumitsoman@cdac.in

Peer-review under responsibility of organizing committee of the International Conference on Information and Communication Technologies (ICICT 2014)
doi:10.1016/j.procs.2015.02.023

There have been many assistive technology applications based on BCI such as the P300 speller[3], robot control[4], prosthesis control[5], games[6] and use in control of virtual environments[7], among others[8,9]. These systems are based on a variety of control paradigms and cater to different requirements of users[10]. However, the BCI system presented here, which is an extension of our work published previously[11], is significant for the following reasons. Firstly, conventional BCI systems require training, both for the system (in terms of training a learning model on a large data corpus) as well as the user (development of the skill to use a BCI system efficiently)[12,13]. However, our system uses eye-blinks as a control intent which is similar for all users, hence the use of the system does not require individual user training by the users. Further, it uses the Emotiv headset which is easy to wear and portable in comparison to other EEG acquisition systems. The integration with a "text-to-speech" system provides a more natural context to the usage of a BCI system, as it suitably augments the conventional human perception of speech. Moreover, it is built on a free and open-source stack and can be configured as per the individual user's convenience.

This paper presents the design and development of a BCI system which enables the user to use eye-blinks to select text options on a computer screen, which are then converted to synthesized speech using a Text-to-Speech (TTS) system. The user's eye-blinks are detected by processing EEG signals using feature extraction and classification techniques. The application allows the user to select the sentence which he/she wishes to speak from a list of configured sentences by blinking when the sentence is highlighted. This is useful for patients who cannot verbally communicate and are physically constrained due to locomotive disorders. They can use this to communicate with their caretakers or other people for day-to-day activities such as wanting food, medicines etc. or to be taken to a particular location.

The rest of the paper is organized as follows. Section 2 discusses the basic architecture and building blocks of a BCI system. Our BCI system is presented in section 3, followed by findings from field trials of the system in Section 4. The conclusions and discussion of our work is presented in Section 5.

2. Architecture of a BCI System

The architecture of a BCI system primarily consists of four stages. The first stage is the EEG signal acquisition phase, which is responsible for interfacing with the EEG acquisition device to provide the EEG data stream for further processing. The second stage is the pre-processing and feature extraction unit. Pre-processing involves artifact removal and filtering of EEG data in the frequency band of interest. The feature extraction stage[14] is responsible for extracting features from the EEG data, which represent significant components of the EEG in a lower-dimensional space. The next stage is the classification stage[15], which uses a classifier to identify the intent from the features obtained from EEG data. In a typical BCI system, the classifier is trained by using features from the training EEG dataset (which constitutes the "offline" stage), and during online usage of the system, the trained classifier determines the intent using features computed from the test EEG data. The final component in the pipeline is the end-user application, which translates the intent identified by the previous stage into the action intended to be performed.

Figure 1 illustrates the architecture of our BCI system. We use the Emotiv EPOC headset which provides EEG data through OpenViBE[16], an open source platform for designing BCI systems. Once the EEG signals are obtained, pre-processing and feature extraction are done using band pass filtering and Common Spatial Pattern (CSP) techniques respectively, which are discussed in the subsequent sections. The classification stage uses Linear Discriminant Analysis (LDA) as a classifier, which identifies the intent and sends it to the end-user application through the OpenViBE desktop bridge. The TTS application receives the intent from the classifier and synthesizes the selected text to speech.

Our system is primarily a single-intent based BCI system that enables a user to operate the TTS system by using eye-blinks. The eye-blinks are visibly prominent in EEG signals and therefore can be detected efficiently. They are also universal for all users, and therefore the system does not need to be trained for each user. The use of eye-blinks as control signals is further substantiated by the fact that patients suffering from motor-neuron disorders primarily use eye-blinks to communicate with others[17]. Conventional BCI systems consider eye-blinks as artifacts and use approaches to remove them from EEG signals[18]. We propose the use of eye blinks as control signals owing to their prominence in EEG signals as compared to other paradigms.

We use the Emotiv EPOC Neuroheadset for EEG signal acquisition, which is a portable 14-channel device. It is easy to setup and use as compared to a clinical EEG device[19]. Eye blinks can easily be detected using the Emotiv headset as it has electrodes placed in the frontal regions, namely AF3, AF4, F7 and F8, according to the 10-20 system[20].

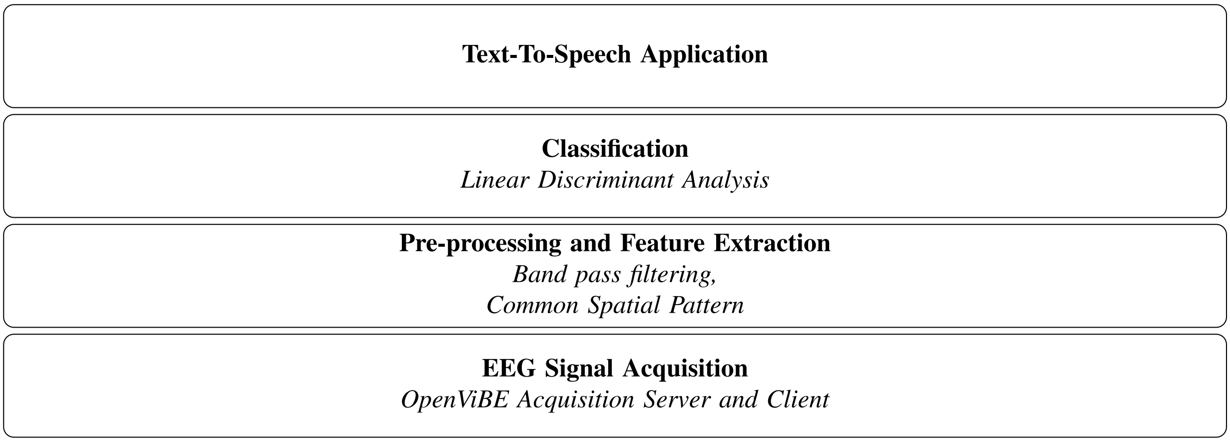

Fig. 1: System Architecture

3. The BCI Processing Pipeline

There are two phases of operation of a BCI System, the offline and online phase. Initially, during both phases, the EEG data of the user is acquired using the intended paradigm and it undergoes pre-processing and feature extraction. During the offline phase, the classifier is trained using the features, whereas during the online phase, it is used to identify the user intent, which is mapped to a command to control a device. Our system is trained once to identify eye blinks, however the system need not be trained subsequently for each user.

This section presents the details of the various components of the BCI processing pipeline mentioned above. We discuss the EEG signal acquisition paradigm in sub-section 3.1, followed by the CSP feature extraction technique in sub-section 3.2. For classification we use the Linear Discriminant Analysis (LDA) which is explained in 3.3, followed by translation to a control command, discussed in sub-section 3.4.

3.1. EEG Signal Acquisition Paradigm

This stage essentially involves acquisition of the EEG signals using a suitable device, such as the Emotiv Neuroheadset. During the offline or training phase, the subject (user) sits facing a computer screen wearing the EEG headset. The EEG of the subject is being continuously recorded and stored in a file. The subject is shown a visual stimulus on the screen, indicating the subject to blink or sit idle. This is termed as an event or a trial. As the subject performs the action intended for the trial, markers are added to the file to enable extraction of EEG signals in a time window around the events for further processing. The user's EEG data is acquired for multiple trials, which constitutes a session. The sequence of trials are randomized to prevent adaptation of the brain[21].

The Emotiv headset filters the data with a low pass filter at 85 Hz cutoff followed by a high pass filter at 0.16 Hz to retain the EEG spectrum. It also applies a notch filter at 50 Hz to remove supply lines interference. As the power spectrum of eye movements is concentrated in the frequency range of 0.5 to 3Hz[22], the EEG is then band passed at 0.5-3 Hz using a Butterworth filter having a pass band ripple of 1dB.

3.2. Feature Extraction using CSP

This section discusses the use of CSP, originally proposed by Ramoser et al.[23], for feature extraction for BCI systems. It has been widely used as it provides better results over other available spatial filters like bipolar filter, Common Average Reference (CAR) filter, Laplacian filter etc[24]. The primary advantage of using CSP is that it addresses the problem of spatial blurring[25].

The steps for implementing CSP are shown in Figure 2. We first compute the covariance matrix of the trials, and then obtain the normalized covariance matrix for data of both classes. These are combined to obtain the composite normalized covariance matrix. Next, we obtain the eigenvalue decomposition of the composite normalized covariance matrix and compute the whitening transformation from the decomposition. We then re-compute the eigenvalue decomposition of the whitened matrix and select the 'k' eigenvectors corresponding to the 'k' highest eigenvalues. Here, 'k' is a configurable parameter of the CSP algorithm. Finally, we project the data to the 'k' dimensional subspace spanned by the selected eigenvectors and take the log-variance of the projected vectors as features for each trial. A detailed description of this algorithm can be found in[23], and its implementation for BCI can be found in our work[11,26].

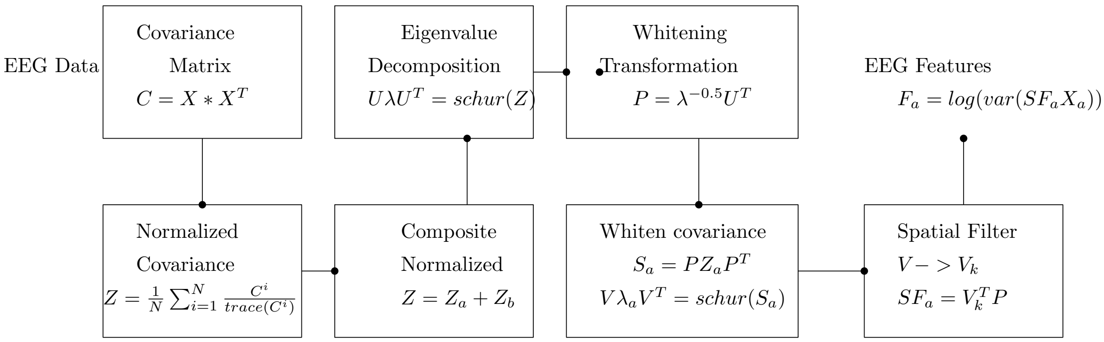

Fig. 2: Flowchart for computing features using Common Spatial Patterns

3.3. Classification using LDA

The next step in the BCI processing pipeline is that of classification, where a classifier is trained to identify the intent from the features. We use the LDA classifier[27], that aims at obtaining a linear decision boundary between the features of the two classes.

The objective of the LDA classifier is to assign a feature to the class having highest conditional probability, thereby minimizing the total error on classification[28]. The classifier is trained using features of the training data, and during online usage of the BCI system, the classifier is used to identify the user intent from the features of the EEG data. In terms of implementation, the output of the classifier during the online phase is sent to an external application that translates the identified intent to a control command or action.

The following are the steps involved in implementing the LDA classifier. Given the matrix of training features X of size $[samples\ X\ features]$ and a label vector Y of size $[samples\ X\ 1]$, where $y_i \in \{0, 1\}$ indicates the class label of the corresponding sample in the feature matrix, we first center the data around the mean for the training data of each class. This is done by subtracting the mean of the training data of each class from each sample of that class.

Next, we compute the covariance matrix from the training data as shown in Equations (1)-(2).

$$\text{Covariance matrix of class i, } c_i = \frac{x_i^T x_i}{n_i} \tag{1}$$

$$\text{Group covariance matrix } C(r, s) = \frac{1}{n}\sum_{i=1}^{g} c_i(r, s) \tag{2}$$

where x_i is the mean centered training sample of class i.

We also compute the prior probability p_i of class i as the ratio of the number of samples of class i to the total number of training samples available for all classes. Further, if the mean for a class of data be denoted by μ_i, then the discriminant function for that class can be computed as shown in Equation (3). We assign an object x_k to a class i that has maximum f_i.

$$f_i = \mu_i C^{-1} x_k^T - \frac{1}{2}\mu_i C^{-1}\mu_i^T + ln(p_i) \tag{3}$$

3.4. End-User Application

A Java application has been developed which takes as input the output of the classifier as discussed in the previous section and provides as output an audio or voice message. The input to the application is binary in nature i.e. it is either a zero or one. Appearance of one indicates the detection of an eye blink made by the user and zero otherwise. The user is shown a panel containing some sentences arranged one below the other which are flashed sequentially. To choose a text, the user has to blink when the desired text is focused and then the text will then be converted to an audio output. As discussed previously, this system will be helpful for patients as the application can be used as a warning system in case the patient needs urgent attention.

The application has been built on a software stack which includes EEGLAB [29] (a MATLAB plugin for EEG signal processing), OpenViBE [16], Java and Emotiv. OpenViBE provides a driver to interface and capture raw EEG signals provided by the Emotiv headset. We developed an OpenViBE scenario that does pre-processing on the signals by passing them through a temporal band pass filter of 0.5Hz to 3Hz, which is the frequency band of interest for detecting eye blinks. These are then passed on to EEGLAB, which realizes the channel selection, feature extraction and classification procedures. The classification result is given back to OpenViBE which then forwards it to the Java application using the VRPN server.

The GUI for the application has been built using Java Swing [30]. At first, the user is shown a screen as in Figure 3a which provides an indication as to whether the system has stabilized or not. During this time the user has to stay calm without doing any muscular movements. This is provided to allow the user some time to get ready to use the system and to prevent the system from providing unexpected outputs while the user may require to adjust the headset position. Once the stability has been achieved, the application automatically takes the user to another screen where texts are flashed sequentially to the user as shown in Figure 3b. The user has to indicate an option by blinking at a time when the desired option is in focus and that option will be "read aloud" by the system. The options are configurable and can be fed to the system using an XML file.

The options can also be multilevel in nature. For example the user can first choose "Help Me" from the rest of options as provided in Figure 3b. Seeing this the application asks the user whether he or she needs an "Attendant" or a "Doctor" for help, as shown in Figure 3c. The application uses three consecutive blinks made by the user as an intent. However, this can also be made configurable depending on the user's comfort level. The application has a text to speech interface which enables the conversion of the text inputs to audio signals. The authors chose to use FreeTTS [31] for speech synthesis as it is written in Java and could be easily integrated with our system. For serializing XML to Java objects a java library called XStream [32] has been used. This became quite useful for implementing the multilevel use of options in the system as discussed before.

(a) Stabilizer Panel (b) Menu (Level 1) (c) Menu (Level 2)

Fig. 3: Screen shots of the application

4. Findings from Field Trials

The developed application was deployed for trials by patients suffering from motor disabilities. The initial training of the system was done by collecting EEG data of 5 subjects with eye-blinks at marked intervals (the data had recordings for two sessions per subject, with 30 trials per session). EEG signals corresponding to rest and eye-blink states were extracted and band-pass filtered in the range of interest (0.5 to 3Hz). CSP features were then extracted from the filtered data and used to train a classifier offline. The system gave an offline accuracy of the order of 95%, averaged across the users. This indicated that the system was suitable for online usage.

As part of performing field trials, the system was deployed at the Neurology wing of a medical facility for use by disabled patients. It was found that users were able to successfully use the application to generate the desired speech from the available options. During the trials, it was observed that the user might also blink involuntarily, which the system may detect as an intent to perform an operation. To address this, a threshold was set to the number of eye blinks to be detected prior to initiation of an operation. For instance, three eye blinks would be required to perform the operation of translating the selected text to speech.

The users gradually became adept at using the application. Initially, most users took some time to understand the operation of the system, primarily with regard to aspects such as the rate of blinking and time at which the user needed to blink. However, with gradual usage, users were able to adapt to using the application comfortably and communicate with their caretakers.

5. Conclusion and Future Work

This paper presented a mechanism to detect eye blinks from EEG signals to use a "Text-to-Speech" application. The accuracy of the system during online classification is appreciably high to enable its day-to-day use. The system has been found to be beneficial for use by patients suffering from motor disabilities as it enables them to communicate with their caretakers or indicate when they need urgent attention, without the prior requirement of training the system.

Future work lies in evolving the processing pipeline to identify imagined movements. The key challenge in developing such a system lies in the inter-session and inter-subject variability in EEG signals for motor imagery. This is presently addressed by training the system for individual users. We intend to evolve such systems to subject-independent scenarios where the system can be used without extensive training.

Acknowledgements

The authors wish to acknowledge the contribution of Soumya Sen Gupta and P Govind Raj for the development and implementation of the BCI system presented in this paper. The authors are also grateful for the efforts of Prof Rupam Borgohain and his team at the Department of Neurology, Nizam's Institute of Medical Sciences, Hyderabad, India for their inputs.

References

1. Lotte, F.. *Study of electroencephalographic signal processing and classification techniques towards the use of brain-computer interfaces in virtual reality applications*. Ph.D. thesis; INSA de Rennes; 2008.
2. Cincotti, F., Mattia, D., Aloise, F., Bufalari, S., Schalk, G., Oriolo, G., et al. Non-invasive brain–computer interface system: towards its application as assistive technology. *Brain research bulletin* 2008;**75**(6):796–803.
3. Farwell, L.A., Donchin, E.. Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and clinical Neurophysiology* 1988;**70**(6):510–523.
4. Millan, J.R., Renkens, F., Mouriño, J., Gerstner, W.. Noninvasive brain-actuated control of a mobile robot by human eeg. *Biomedical Engineering, IEEE Transactions on* 2004;**51**(6):1026–1033.
5. Müller-Putz, G.R., Scherer, R., Pfurtscheller, G., Rupp, R.. Eeg-based neuroprosthesis control: a step towards clinical practice. *Neuroscience letters* 2005;**382**(1):169–174.
6. Nijholt, A., Tan, D., Allison, B., del R Milan, J., Graimann, B.. Brain-computer interfaces for hci and games. In: *CHI'08 extended abstracts on Human factors in computing systems*. ACM; 2008, p. 3925–3928.
7. Faller, J., Müller-Putz, G., Schmalstieg, D., Pfurtscheller, G.. An application framework for controlling an avatar in a desktop-based virtual environment via a software ssvep brain-computer interface. *Presence: Teleoperators and Virtual Environments* 2010;**19**(1):25–34.

8. Millán, J.d.R., Rupp, R., Müller-Putz, G.R., Murray-Smith, R., Giugliemma, C., Tangermann, M., et al. Combining brain–computer interfaces and assistive technologies: state-of-the-art and challenges. *Frontiers in neuroscience* 2010;**4**.
9. Zickler, C., Riccio, A., Leotta, F., Hillian-Tress, S., Halder, S., Holz, E., et al. A brain-computer interface as input channel for a standard assistive technology software. *Clinical EEG and Neuroscience* 2011;**42**(4):236–244.
10. Curran, E.A., Stokes, M.J.. Learning to control brain activity: a review of the production and control of eeg components for driving brain–computer interface (bci) systems. *Brain and cognition* 2003;**51**(3):326–336.
11. Gupta, S., Soman, S., Raj, P., Prakash, R., Sailaja, S., Borgohain, R.. Detecting eye movements in eeg for controlling devices. In: *Computational Intelligence and Cybernetics (CyberneticsCom), 2012 IEEE International Conference on*. 2012, p. 69–73. doi:`10.1109/CyberneticsCom.2012.6381619`.
12. Allison, B., Luth, T., Valbuena, D., Teymourian, A., Volosyak, I., Graser, A.. Bci demographics: How many (and what kinds of) people can use an ssvep bci? *Neural Systems and Rehabilitation Engineering, IEEE Transactions on* 2010;**18**(2):107–116.
13. Guger, C., Daban, S., Sellers, E., Holzner, C., Krausz, G., Carabalona, R., et al. How many people are able to control a p300-based brain–computer interface (bci)? *Neuroscience letters* 2009;**462**(1):94–98.
14. McFarland, D.J., Anderson, C.W., Muller, K., Schlogl, A., Krusienski, D.J.. Bci meeting 2005-workshop on bci signal processing: feature extraction and translation. *IEEE transactions on neural systems and rehabilitation engineering* 2006;**14**(2):135.
15. Lotte, F., Congedo, M., Lécuyer, A., Lamarche, F., Arnaldi, B., et al. A review of classification algorithms for eeg-based brain–computer interfaces. *Journal of neural engineering* 2007;**4**.
16. Renard, Y., Lotte, F., Gibert, G., Congedo, M., Maby, E., Delannoy, V., et al. Openvibe: an open-source software platform to design, test, and use brain-computer interfaces in real and virtual environments. *Presence: teleoperators and virtual environments* 2010;**19**(1):35–53.
17. Bauer, G., Gerstenbrand, F., Rumpl, E.. Varieties of the locked-in syndrome. *Journal of neurology* 1979;**221**(2):77–91.
18. Joyce, C., Gorodnitsky, I., Kutas, M.. Automatic removal of eye movement and blink artifacts from eeg data using blind component separation. *Psychophysiology* 2003;**41**(2):313–325.
19. Lievesley, R., Wozencroft, M., Ewins, D.. The emotiv epoc neuroheadset: an inexpensive method of controlling assistive technologies using facial expressions and thoughts? *Journal of Assistive Technologies* 2011;**5**(2):67–82.
20. Homan, R., Herman, J., Purdy, P.. Cerebral location of international 10–20 system electrode placement. *Electroencephalography and clinical neurophysiology* 1987;**66**(4):376–382.
21. Guger, C., Edlinger, G., Harkam, W., Niedermayer, I., Pfurtscheller, G.. How many people are able to operate an eeg-based brain-computer interface (bci)? *Neural Systems and Rehabilitation Engineering, IEEE Transactions on* 2003;**11**(2):145–147.
22. Manoilov, P.. Eeg eye-blinking artifacts power spectrum analysis. In: *Proceedings of International Conference on Computer Systems and Technologies*. 2006, p. 3–5.
23. Ramoser, H., Muller-Gerking, J., Pfurtscheller, G.. Optimal spatial filtering of single trial eeg during imagined hand movement. *Rehabilitation Engineering, IEEE Transactions on* 2000;**8**(4):441–446.
24. Dornhege, G., Millán, J., Hinterberger, T., McFarland, D., Müller, K.. *Toward brain-computer interfacing*; vol. 74. MIT press Cambridge, MA; 2007.
25. Napoli, A., Obeid, I.. Combined common spatial pattern and spectral filtering for eeg-based bcis. In: *Neural Engineering (NER), 2011 5th International IEEE/EMBS Conference on*. IEEE; 2011, p. 449–452.
26. Gupta, S., Soman, S., Raj, P., Prakash, R.. Improved classification of motor imagery datasets for bci by using approximate entropy and wosf features. In: *Signal Processing and Integrated Networks (SPIN), 2014 International Conference on*. 2014, p. 90–94. doi:`10.1109/SPIN.2014.6776928`.
27. Muller, K., Mika, S., Ratsch, G., Tsuda, K., Scholkopf, B.. An introduction to kernel-based learning algorithms. *IEEE transactions on neural networks* 2001;**12**(2):181–201.
28. Teknomo, K.. Discriminant analysis tutorial. 2006. [Online; accessed 26-October-2012].
29. Delorme, A., Makeig, S.. Eeglab: an open source toolbox for analysis of single-trial eeg dynamics including independent component analysis. *Journal of neuroscience methods* 2004;**134**(1):9–21.
30. Eckstein, R., Loy, M., Wood, D.. *Java swing*. O'Reilly & Associates, Inc.; 1998.
31. Walker, W., Lamere, P., Kwok, P.. Freetts: a performance case study 2002;.
32. Girod, L., Mei, Y., Newton, R., Rost, S., Thiagarajan, A., Balakrishnan, H., et al. Xstream: A signal-oriented data stream management system. In: *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*. IEEE; 2008, p. 1180–1189.