Multiresolution analysis over graphs for a motor imagery based online BCI game

Javier Asensio-Cubero [a,*], John Q. Gan [a], Ramaswamy Palaniappan [b]

[a] University of Essex, Wivenhoe Park, Colchester, Essex CO4 3SQ, United Kingdom
[b] School of Computing, University of Kent, Chatham Maritime, Kent ME4 4AG, United Kingdom

ABSTRACT

Multiresolution analysis (MRA) over graph representation of EEG data has proved to be a promising method for offline brain–computer interfacing (BCI) data analysis. For the first time we aim to prove the feasibility of the graph lifting transform in an online BCI system. Instead of developing a pointer device or a wheel-chair controller as test bed for human–machine interaction, we have designed and developed an engaging game which can be controlled by means of imaginary limb movements. Some modifications to the existing MRA analysis over graphs for BCI have also been proposed, such as the use of common spatial patterns for feature extraction at the different levels of decomposition, and sequential floating forward search as a best basis selection technique. In the online game experiment we obtained for three classes an average classification rate of 63.0% for fourteen naive subjects. The application of a best basis selection method helps significantly decrease the computing resources needed. The present study allows us to further understand and assess the benefits of the use of tailored wavelet analysis for processing motor imagery data and contributes to the further development of BCI for gaming purposes.

© 2015 Published by Elsevier Ltd.

1. Introduction

During the recent years many studies have focused on the use of electroencephalographic data (EEG) for human–machine interaction. This paradigm, known as brain–computer interfacing (BCI), is grounded on a diverse range of disciplines such as neuroscience, machine learning and digital signal processing among others.

The classification of imaginary limb movements has proven to be an adequate approach for augmenting motor functions for disabled and healthy subjects [1–4]. The physical basis of motor imagery (MI) BCIs comes from the changes on the μ rhythm during the performance of MI tasks, which is known as event-related desynchronisation (ERD) and event-related synchronisation (ERS) [5].

These changes on the EEG data occur in different locations on the scalp, at different time instants and on different frequencies. EEG data is also known to be highly noisy, and the patterns arisen during the MI process drastically change among different subjects. These characteristics make the analysis of MI data a remarkable complex task.

Wavelet analysis has been profusely applied for the analysis of EEG data [6–8]. The characteristics of this orthogonal system presents important benefits as it offers temporal-spectral analysis along different resolution levels. The introduction of the second generation wavelets has leveraged the design of new wavelet families that can adapt better to the domain of study [9].

In the present work we aim to explore the feasibility of applying multiresolution analysis over EEG data graph representation for an online real-time BCI system. The graph representation allows to embed the spatial information during the multiresolution analysis process covering the three dimensions involved in the ERS/ERD development (temporal, spectral and spatial dimensions). The method, fully described in [10], introduces the concept of tailored wavelet lifting for brain–computer interfaces.

The proposed online system is an *endless running game* where the subject has to control a character while it is constantly running forwards. During the game play the subject will have to decide which command to send to the game (either jump, stride left or stride right) depending on the game state at a given time. Video games have been recently used in the BCI field as they are easier to implement than other direct applications such as BCI controlled wheelchairs or robotic arms, and more engaging than spellers or pointing devices [11,12]. A detailed state of the art of the use of games in the BCI field is given in [13].
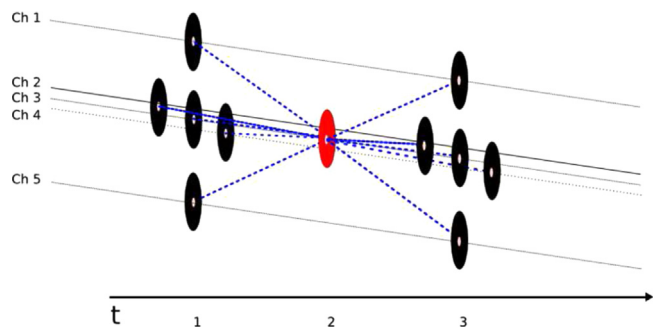
This paper is structured as follows. The data acquisition and preprocessing are detailed in Section 2.1.1. In Section 2.1.2 wavelet lifting on graphs is described. Section 2.1.3 describes the feature

* Corresponding author. Tel.: +44 7587223814.
E-mail addresses: capitan.cambio@gmail.com (J. Asensio-Cubero),
jqgan@essex.ac.uk (J.Q. Gan), r.palani@kent.ac.uk (R. Palaniappan).

Fig. 1. Details of the graph after the even/odd split. Even nodes (black circles) are used to compute the detail coefficients, approximating the odd nodes (red circles). The number of channels depicted has been reduced for clarity, during the experiments 15 channels were used [10]. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

extraction technique applied and the classification method is detailed in Section 2.2. The game design along with the acquisition protocol are presented in Section 2.3. The results and discussions are detailed in Section 3 and conclusions are drawn in Section 4.

2. Methods

2.1. Data analysis and feature extraction

2.1.1. Data acquisition and preprocessing

Three different imaginary movements (right hand, left hand and feet) were recorded from fourteen healthy subjects, all of them naive on the use of online BCI. The subjects aged from 24 to 32 and 50% were female. All the participants, recruited from different schools and faculties at University of Essex, signed a consent form where the details of the experiment and the use of the acquired data were explained.

The data was recorded with a sampling frequency of 256 Hz. The monopolar electrodes covered the major part of the cortex area, specifically the following 15 locations: Fc3, Fc1, FcZ, Fc2, Fc4, C3, C1, CZ, C2, C4, Fp3, Fp1, FpZ, Fp2 and Fp4. Two reference electrodes were placed under the subjects' ears. The data from these two electrodes were averaged and subtracted from the rest of the electrodes at every sampling point.

Eight seconds of data were recorded for each trial and filtered using an elliptic band pass filter between 8 and 30 Hz. The signals were then cropped from $t = 2$ s to the instant $t = 7$ s as this period of the trial provides adequate information for motor imagery classification. Finally, a sliding window of one second with a sliding step of 50 was applied until the signal was divided into 20 segments. No extra preprocessing step was carried out in terms of artifact removal in order to assure a more dynamic feedback to the subject. In Section 2.3 a detailed explanation of the acquisition protocol is given.

The recording hardware used was the *BioSemi*'s *ActiveTwo* that reads the EEG signals using active electrodes.

2.1.2. Lifting transform over graphs

The data analysis is based on a graph lifting transform [14,15] over EEG data graph representation. Each MI segment of T samples and C channels $X^{T \times C}$ is embedded in a graph $G = (V, E)$, where V is the vertex set (a flattened version of X) and E represents the graph edges.

The edges of G are arranged such that they capture the temporal and spatial relationships present in the data segment X and they are represented by using an adjacency matrix Adj. As shown in Fig. 1, the node $v_{c_3,t}$, located on channel c_3 at instant t, is linked

to $v_{c_3,t-1}$ and $v_{c_3,t+1}$ which are the closest temporal neighbours in the same channel. With the graph representation we also provide spatial information by including the four neighbouring electrodes of $v_{c_3,t-1}$: $v_{c_1,t-1}$, $v_{c_2,t-1}$, $v_{c_4,t-1}$ and $v_{c_5,t-1}$; and four more neighbours of $v_{c_3,t+1}$: $v_{c_1,t+1}$, $v_{c_2,t+1}$, $v_{c_4,t+1}$ and $v_{c_5,t+1}$. It is noteworthy that the proposed graph architecture makes the application of a graph lifting transform straight forward.

As in any lifting transform we need to define the *split, predict* and *update* steps.

The *split* step is defined over the node set by using the parity of t. The even vertex set V_e corresponds to the elements in segment X at even values of t, and analogously, the odd set V_o corresponds to the elements at odd values of t.

Prior to the definition of the *predict* and *update* functions the vertex set V of size $N = N_o + N_e$ has to be rearranged. The odd vertices V_o of size $N_o \times 1$ are relocated preceding the even vertices V_e of size $N_e \times 1$ obtaining the following graph definition:

$$\tilde{V} = \begin{pmatrix} V_o \\ V_e \end{pmatrix}$$

$$\tilde{Adj} = \begin{pmatrix} F^{N_o \times N_o} & J^{N_o \times N_e} \\ K^{N_e \times N_o} & L^{N_e \times N_e} \end{pmatrix} \quad (1)$$

The submatrices F and L in $\tilde{Adj}$ in Eq. (1) link the elements within the same node sets and are empty, so they are discarded. The block matrix J contains only edges linking odd elements to even elements and, analogously, K only links even elements to odd elements.

The lifting analysis function is then defined as

$$D = V_o - J^\omega \times V_e$$
$$A = V_e + K^\omega \times D \quad (2)$$

In Eq. (2) the prediction and update functions are defined as the matrix product $\mathcal{P} = J^\omega \times V_e$ and $\mathcal{U} = K^\omega \times D$, where J^ω and K^ω are the weighted adjacency block matrices. The predict matrix J is weighted row-wise applying the equation $J^\omega_{i,j} = 1/(\sum_{j'} J_{i,j'})$ for each row i and column j, j' is the index used to iterate through the columns. The weighted version of K is analogously computed as $K^\omega_{i,j} = 1/2 * (\sum_{j'} J_{i,j'})$. The weighting is performed in order to maintain the spatio-temporal properties of the original graph and has been designed based on the linear wavelet lifting [10].

As a result of the lifting transform we obtain the detail coefficient set D and the approximation coefficient set A. The process described by Eq. (2) is repeated in each level. For level $l+1$, V is set as A obtained in level l. For the present study the transform was calculated for the first five levels of decomposition, obtaining ten coefficient sets $\{D^l, A^l\}, l \in 1, 2, 3, 4, 5$.

2.1.3. Common spatial patterns

The coefficient sets resulted from the transform in each level l, A^l and D^l, belong to $\mathbb{R}^{(T/2^l) \times C}$ with $T = 256$ and $C = 15$. Therefore, given the size of the data, a feature extraction step is needed before the classification process. For this task we chose to apply common spatial patterns (CSP) [16] as it has been successfully applied in other studies with a similar design [17,18].

For ease of understanding, we refer to the detail D^l and approximation A^l sets at different levels as $\overline{X}$. Each coefficient set is projected onto its own CSP space $\overline{Y} = W^T \times \overline{X}$ where W^T is the transposed CSP projection matrix.

CSP is a supervised spatial filtering technique which maximises the variance ratio between two different classes. It is computed as the generalised eigenvector decomposition of the estimated covariance matrix $\Sigma^{(+)}$ of the trials belonging to class $(+)$, and the estimated covariance matrix for the trials belonging to class (-), $\Sigma^{(-)}$:

$$\Sigma^{(+)} = W \Lambda^{(+)} W^T \quad$$

$$\Sigma^{(-)} = W\Lambda^{(-)}W^T \qquad (3)$$

In Eq. (3) $\Lambda^{(+)}$ and $\Lambda^{(-)}$ are diagonal matrices with the eigenvalues corresponding to the decomposition of $\Sigma^{(+)}$ and $\Sigma^{(-)}$. A large eigenvalue $\Lambda^{(+)}_{jj}$ implies that the corresponding eigenvector from matrix W, $\mathbf{w}_j$, leads to high variance in the projected signal in the positive class and low variance in the negative one (and vice-versa).

As indicated in [16], we computed the variances of the first m rows and last m rows from $\overline{Y}$ as $f_k = var(\overline{\mathbf{y}}_k)$ with $k = 1, 2, \ldots, m, C-(m-1), \ldots, C$. The final features for the classification are scaled using the logarithm $f_k^{log} = \log(f_k / \sum_{j=1}^{F} f_j)$.

2.2. Classification and best basis selection

Due to the large amount of features produced during the signal analysis step (20 segments and 10 coefficient sets per segment) we have to restrict the possible classification techniques to one that is fast enough and does not need extra tuning to achieve competitive results. For this purpose we should not take into consideration artificial neural networks as they would require a considerable amount of time to train. Support vector machines, although they have proved to obtain high classification performance in the field, usually need some form of parameter adjustment via cross-validation. For these reasons the most appropriate classification method would be linear discriminant analysis (LDA) [19], as it is fast, there is no need to temper any parameters, and has proved to achieve good classification rates in the field.

The features from each level, each coefficient set (detail or approximation), and each temporal segment were classified with a separate linear discriminant model. Therefore, for every trial to be classified, we obtained a total of $N_{lda} = N_s * l * 2$ LDA outputs, with N_s being the number of segments and l the number of levels.

In [10] it is discussed how in a similar setup the classification accuracy could be enhanced by choosing appropriate segments and levels when computing the majority voting prior to the final classification decision for a trial. For this goal we applied sequential floating forward search (SFFS) [20] over the cross-validation LDA outputs from the training data. Although SFFS is commonly applied for feature selection this approach would slow down considerably the training process, the philosophy of the algorithm allows us to use it as a *best bases selection* of the classification outputs, which takes less time than feature selection.

SFFS is a bottom-up procedure which dynamically changes the number of selected features (in our case bases) by discarding those which do not contribute to maximise a given cost function, and adding those which improve the output.

Let us assume that we have a total of N_{lda} LDA outputs $O = o_{j_{i=1}^{N_{lda}}}$ and we have already selected a subset of them P_k with the corresponding cost function $J(P_k)$. For this study the cost function is defined as the median of the Kappa values over the five cross-validation folds using P_k for the majority voting. The Kappa value was computed as $\kappa = p_o - p_c / 1 - p_c$, where p_o is the proportion of units on which the judgement agrees (based on the output from the classifier and the actual label), and p_c is the proportion of units on which the agreement is expected by chance [21].

The SFFS algorithm steps are

1. *Inclusion*: Select the next feature p_{k+1} to include in the feature set from the remaining candidates $O - P_k$, based on the cost function $J(P_{k+1})$ value where $P_{k+1} = P_k + p_{k+1}$.
2. *Condition exclusion*: Exclude the least meaningful feature from P_{k+1}.
   - Find the least significant feature p_r by $J(P_{k+1} - p_r) \leq J(P_k), \forall r \in 1, 2, \ldots, k$ and set $P'_k = P_{k+1} - p_r$. If $k = 2$ set $P_k = P'_k$ and return to 1; otherwise go to 3.

3. *Continuation of conditional exclusion*: Find the least significant feature p_s in P'_k.
   - If $J(P'_k - p_s) > J(P_{k-1})$ then set $P_k = P'_k$ and return to 1.
   - Else $J(P'_k - p_s) \leq J(P_{k-1})$, If $k = 2$ update $P_k = P'_k - p_s$ and return to 1; else repeat 3 with $P'_k = P'_k - p_s$.

The algorithm is initialised with $k = 0$, $P_0 = O$ and no restriction of maximum number of features to select was imposed.

2.3. Game and protocol design

In order to evaluate the lifting scheme over graphs as the signal processing method for online brain–computer interfaces a simple game called *brainz* was developed. The goal of this game is to help the main character to catch as many coins as possible and to avoid being bitten by the snakes. The game play used is known as *running game*, where the player character is seen from a distance and keeps moving forwards during the playing time. The game view consists of three lanes and the player character is placed in the middle one as shown in Fig. 2. At the beginning of each trial either one coin or three snakes start to approach from the horizon towards the player character. By the end of the trial and depending on the classifier decision, one of the following action is performed: moving towards the right or left lane, by performing right hand or left hand imagery movements, in order to fetch the coin (and then coming back to the central lane), or jumping over the snakes by performing feet imagery movements. At the end of the trial the character comes back to the central lane automatically. If the classifier decision is coherent with the interface state (coin approaching from one of the sides or snakes coming down from the three lanes) the subject scores 100 points. In each run the three classes were evenly distributed and the order in which each class appeared was randomised.

Complementing the game interface, there is also an MI state interface at the bottom of the screen (See Fig. 3). The trial duration is 8 s. At $t = 0$ s a fixation cross is displayed, after two seconds ($t = 2$ s) the fixation cross disappears and three arrows are shown (pointing left, right and up) at the same time when an aural cue is played. One of the arrows is highlighted by a blue border indicating which MI to perform (left hand, right hand and feet). At $t = 8$ s the arrow with the blue border turns green indicating the end of the trial. A resting period of two seconds was given to the subject, after which the fixation cross was displayed again. Both interfaces, the game and the BCI state, are fully synchronised.

During the experiment each subject performed four runs of 30 trials each, in order to gather data for calibration, obtaining a total of 40 trials per class. In each run the subject was cued to perform ten trials of each class, the order in which each class shown was randomised. This produced an even distribution of classes per run.

During the calibration stage the game played by itself letting the subject familiarise with the game interface while recording the MI data.

After the calibration phase the BCI system was trained, which was performed for each of the subjects separately. The number of CSP features to use (one, two or three) and the best bases were selected by applying 5-fold cross-validation in order to speed up the training process.

The online game play was identical to the calibration phase except for two major changes. Firstly, the character reacted driven by the classifier's output to the subjects' signals, and secondly, every fourth of a second one of the arrows was coloured in red, giving feedback of the intermediate classifier output. Each subject performed three runs, 30 trials in each run, using the online BCI game with feedback. After each run the system was recalibrated so that the classifier could cope with the changes derived from the online feedback.
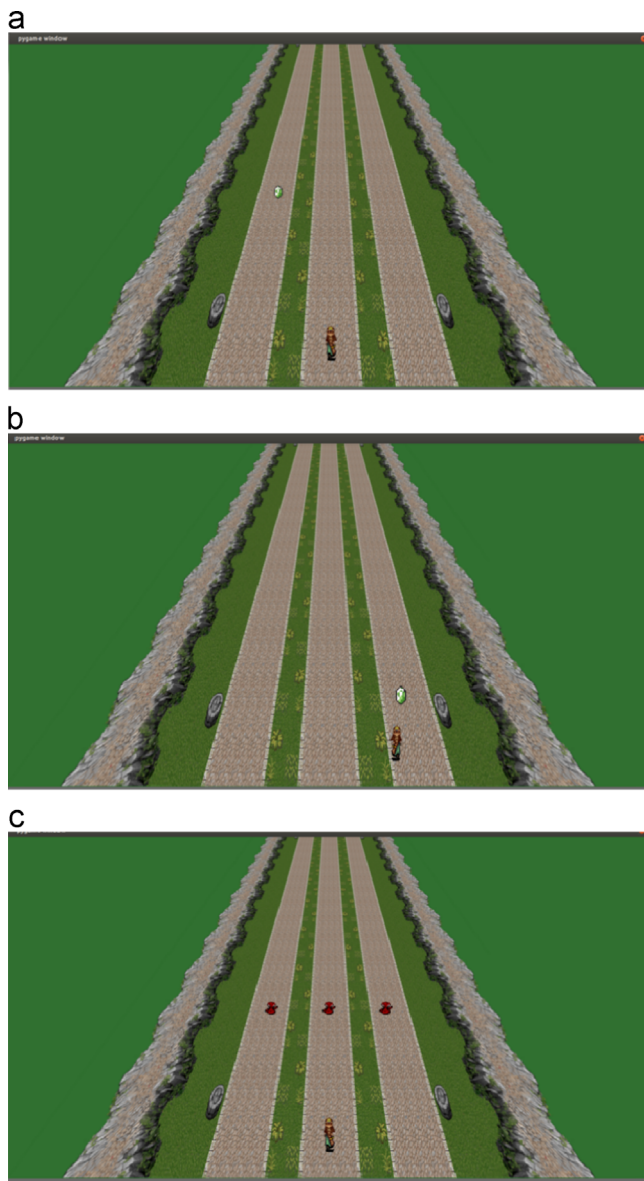
Fig. 2. Three screen shots of the game interface showing the three different target types. (a) Shows a coin coming down from the left lane. The subject is supposed to perform left-hand MI in order to catch it. (b) Shows the playing character moving towards the right lane to grab the coin. (c) Shows the three snakes, one per lane, coming down towards the playing character. The subject has to perform the feet MI in order to trigger the jump action.

The subject was given resting periods of five to ten minutes between runs (or longer if the subject wished so). The whole experiment was performed in within around two hours, taking into account the electrode setup time.

3. Results and discussions

3.1. Classification accuracy

The results in terms of classification accuracies and Kappa values of the different runs are shown in Tables 1–3 respectively. From the results we can observe how the Kappa values (and the classification accuracy) increases run by run. This is due to two different factors. Firstly, the number of trials available for training the model increases after each run, helping to generate better LDA models. Secondly, the
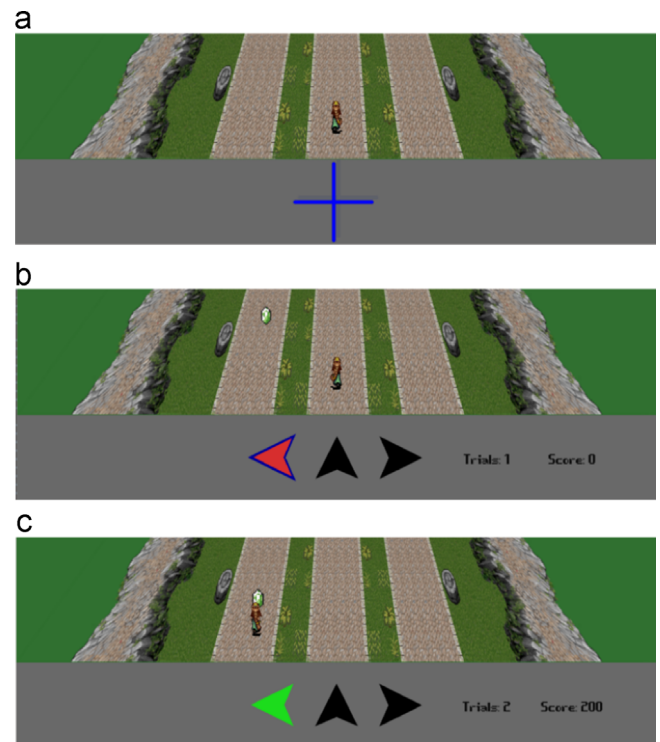
Fig. 3. Details of the MI state interface. (a) Fixation cross alerting the subject that an MI should be performed in two seconds. (b) The arrow marked with the blue border indicates the expected movement to imagine. The inter-trial partial classifications are show to the user by a read arrow. (c) The green arrow indicates the final classifier's decision and the end of the trial. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

Table 1
First online run: Kappa values and classification accuracies. *Training Acc.* and *Training Kappa* stand for the crossvalidation classification rate and Kappa values, *Online Acc.* and *Online Kappa* stand for the online classification rate and Kappa values. The mean values and standard deviation are provided at the bottom of the table.

Subject	Training acc.	Training kappa	Online acc.	Online kappa	Game score
1	0.82	0.72	0.60	0.40	1800
2	0.92	0.87	0.73	0.59	2200
3	0.94	0.91	0.87	0.80	2600
4	0.78	0.66	0.50	0.26	1500
5	0.76	0.64	0.50	0.25	1500
6	0.56	0.34	0.47	0.20	1400
7	0.57	0.34	0.40	0.10	1200
8	0.63	0.45	0.60	0.40	1800
9	0.76	0.65	0.63	0.45	2000
10	0.66	0.47	0.40	0.10	1200
11	0.82	0.72	0.47	0.20	1400
12	0.62	0.45	0.53	0.30	1600
13	0.40	0.11	0.33	0.00	1000
14	0.70	0.55	0.40	0.10	1200
Mean	0.71 ± 0.15	0.56 ± 0.22	0.53 ± 0.15	0.30 ± 0.22	1600 ± 44.20

data gathered during the online game play may contain features not present during the offline data analysis as the subject's EEG patterns could change due to the frustration and stress derived from the real-time BCI control.

This improvement proves to be statistically significant between the first and third runs when using Wilcoxon's ranksum test (used due to the small size of the population and its paired nature), by comparing both Kappa value ($p=0.021$) and classification accuracy ($p=0.020$).

Table 2

Second online run: Kappa values and classification accuracies. *Training Acc.* and *Training Kappa* stand for the crossvalidation classification rate and Kappa values, *Online Acc.* and *Online Kappa* stand for the online classification rate and Kappa values. The mean values and standard deviation are provided at the bottom of the table.

Subject	Training acc.	Training kappa	Online acc.	Online kappa	Game score
1	0.76	0.64	0.70	0.55	2100
2	0.83	0.75	0.80	0.70	2400
3	0.92	0.88	0.87	0.80	2600
4	0.65	0.49	0.53	0.34	1600
5	0.69	0.54	0.50	0.25	1500
6	0.55	0.33	0.43	0.15	1300
7	0.47	0.21	0.47	0.20	1400
8	0.59	0.38	0.67	0.50	2000
9	0.76	0.64	0.60	0.40	1800
10	0.55	0.32	0.57	0.35	1700
11	0.73	0.59	0.53	0.30	1600
12	0.64	0.46	0.43	0.15	1300
13	0.49	0.23	0.57	0.35	1700
14	0.65	0.48	0.63	0.45	2000
Mean	0.66	0.50	0.59	0.39	1785.70
	± 0.13	± 0.19	± 0.13	± 0.19	± 394.00

Table 3

Third online run: Kappa values and classification accuracies. *Training Acc.* and *Training Kappa* stand for the crossvalidation classification rate and Kappa values, *Online Acc.* and *Online Kappa* stand for the online classification rate and Kappa values. The mean values and standard deviation are provided at the bottom of the table.

Subject	Training acc.	Training kappa	Online acc.	Online kappa	Game score
1	0.77	0.65	0.57	0.35	1700
2	0.89	0.83	0.63	0.49	2000
3	0.89	0.83	0.83	0.75	2500
4	0.66	0.50	0.77	0.65	2300
5	0.70	0.55	0.57	0.35	1700
6	0.63	0.45	0.50	0.25	1500
7	0.56	0.34	0.43	0.15	1300
8	0.72	0.58	0.77	0.65	2300
9	0.80	0.69	0.70	0.55	2100
10	0.62	0.42	0.53	0.30	1600
11	0.76	0.63	0.70	0.55	2100
12	0.61	0.41	0.67	0.50	2000
13	0.56	0.34	0.50	0.25	1500
14	0.76	0.64	0.70	0.55	2100
Mean	0.71	0.56	0.63	0.45	1900
	± 0.11	± 0.16	± 0.12	± 0.18	± 358.30

This difference is not significant if we compare the results between the first and second runs (Kappa value $p = 0.090$, classification rate $p = 0.094$); and the second and third runs (Kappa value $p = 0.178$, classification rate $p = 0.177$). There were some subjects though, such as *Subject 1*, who actually obtained lower performance during the last game play.

3.2. Best basis selection outcomes

The use of a best basis selection method prior to the game play has two different motivations. On one hand, best basis selection methods often help to improve the classification accuracy. On the other hand, they reduce the amount of data to process during the online analysis (Fig. 4).

In Fig. 5 different classification rates and Kappa values for the online game play are compared without applying SFFS against applying SFFS. It is noteworthy that the classification accuracies and Kappa values in the first two runs are not improved by using SFFS, although this difference is not statistically significant. We believe that
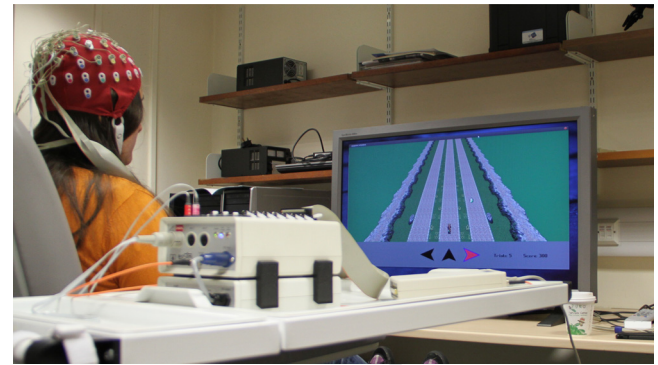

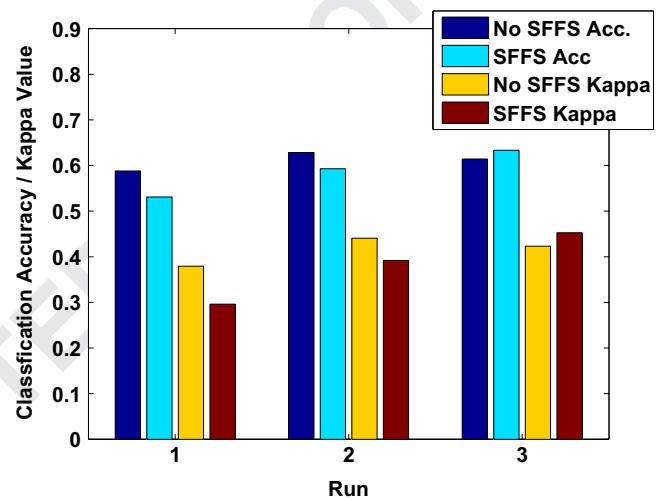
Fig. 4. Subject during the online game play.

Fig. 5. Classification accuracies and Kappa values for three game play runs without SFFS against using SFFS for best basis selection.

this is because the use of SFFS introduces a slight overfitting in the classification process.

In terms of reducing the amount of data to be processed SFFS proves to be an efficient method. The initial 200 coefficient sets were reduced on average (including the standard deviation) per run to: 81.35 ± 10.81 (first run), 92.85 ± 7.70 (second run), and 78.07 ± 12.56 (third run). After applying SFFS we would decrease the number of CSP and LDA evaluations to a half in the worst case and to 84.09 ± 12.12 on average. The process of sequentially preprocessing, computing the multiresolution analysis, feature extraction, classification and majority voting adds up to 1.66 ± 0.026 s. Taking into account that applying CSP and classifying the features takes 0.004 ± 0.0004 s for each of the 200 coefficient sets, the total time spent in this task per trial is 0.8403 ± 0.006 s. Therefore, when applying SFFS we obtained an average speed-up factor of 1.41.

Although it is an implementation detail, we need to point out that the online system was completely paralellised based on segments. Thus, each segment was processed and its classification output computed as soon as there was enough data to do so, which means that the latency that we had once the trial was finished was approximately 80 ms for the worst case. Therefore, the response time we got from the online system is low enough and suitable for real-time applications.

The overfitting issue in applying SFFS is more notorious in the first online run, where the SFFS approach resulted in an average Kappa value of 0.29 ± 0.21 and the non-SFFS approach 0.37 ± 0.23. The reason for this difference is that, as we already pointed out, the features from the dataset acquired during the calibration process may have some differences with the features produced during the game play, provoked mainly by changes in the subject's mental state, namely

frustration and stress. SFFS selects the theoretically optimal coefficient sets (according to the heuristic provided) to take into account for the trial classification which, specially in the first run, differ from the real optimal set. Therefore taking into account more coefficient sets may lead to better Kappa values when there are big differences in the training and the evaluation sets. This notion was validated by the fact that there exists a correlation ($p=0.053$) between the number of coefficient sets used by SFFS and the Kappa value achieved only in the results of the first run. In the following runs, as the data gathered during the game play was added to the training pool, this correlation vanished.

4. Conclusions

In this study we have applied a tailored wavelet transform over graphs for a real time BCI for the first time. One of the most important issues to overcome when designing an online BCI application is the need of a real-time response from the system. The wavelet lifting transform is less resource consuming in terms of memory and computation than first generation wavelets [10], and has proven to meet the requirements for this study.

The proposed methodology achieved a classification accuracy of 0.63 ± 0.12 and a Kappa value of 0.45 ± 0.18, which is satisfactory for a 3-class BCI online system with 14 naive subjects, specially for the relative small training set. If we applied the proposed method to a similar offline data set, with a similar protocol [22] we obtained a classification rate of 0.72 ± 0.14 and a Kappa value of 0.58 ± 0.21. We observed that there was a drop 0.09 in the classification accuracy from offline to online, which is a common difference between offline and online BCI systems.

The use of SFFS in the current configuration has helped us to significantly reduce the number of coefficient sets to be projected onto their CSP spaces and to calculate their LDA outputs, reducing the computer power needed when using our system in a real time context, offering an average speed up of 1.41 times for each trial evaluated during the game play. Anyhow, it is convenient to remember that we should design BCI online systems able to cope with the theoretical worst case, the one where all the coefficient sets are selected by the SFFS algorithm. It was observed that in the practical worst case the number of segments to be processed was reduced to a half by SFFS. In terms of classification rate, applying majority voting over the basis selected by SFFS fails to outperform the simple majority voting.

The game paradigm utilised has proved to be convenient for developing and testing BCI online systems. One one hand, in *endless running games* it is easy to determine which is the next appropriate move depending on the state of the game (in contrast with other game paradigms in which this would be more difficult to know), and therefore, measure the classification performance of the system. On the other hand, the game play has been utilised for many years in the game industry and it has proven to be engaging enough for our study. The tendency of slightly improving the score after each run made many of the subjects to express the will to improve their score of the previous runs, and some of them even asked to play extra runs once the experiment was through. A study of the correlation between the subject stress and frustration levels with the classification rate would also give insights of the possible enhancements to BCI gaming systems.

In conclusion, the use of multiresolution analysis over EEG data graph representation produces promising results to online BCI systems. The results presented hereby encourage us to improve the proposed method and to explore the possibilities offered by the lifting transform and the graph representation.

Conflict of interest statement

None declared.

References

[1] Y. Pang, Y. Yuan, K. Wang, Learning optimal spatial filters by discriminant analysis for brain–computer interface, Neurocomputing 77 (1) (2012) 20–27.
[2] J.d.R. Millan, R. Rupp, G.R. Muller-Putz, R. Murray-Smith, C. Giugliemma, M. Tangermann, C. Vidaurre, F. Cincotti, A. Kbler, R. Leeb, Combining brain–computer interfaces and assistive technologies: state-of-the-art and challenges, Front. Neurosci. 4 (2010) 161.
[3] O. Bai, P. Lin, D. Huang, D. Fei, M.K. Floeter, Towards a user-friendly brain–computer interface: initial tests in ALS and PLS patients, Clin. Neurophysiol. 121 (8) (2010) 1293–1303.
[4] G. Onose, C. Grozea, A. Anghelescu, C. Daia, C.J. Sinescu, A.V. Ciurea, T. Spircu, A. Mirea, I. Andone, A. Spnu, On the feasibility of using motor imagery EEG-based brain–computer interface in chronic tetraplegics for assistive robotic arm control: a clinical test and long-term post-trial follow-up, Spinal Cord 50 (8) (2012) 599–608.
[5] G. Pfurtscheller, F.H. Lopes da Silva, Event-related EEG/MEG synchronization and desynchronization: basic principles, Clin. Neurophysiol. 110 (11) (1999) 1842–1857.
[6] B. Perseh, A.R. Sharafat, An efficient p300-based BCI using wavelet features and IBPSO-based channel selection, J. Med. Signals Sensors 2 (3) (2012) 128.
[7] O. Carrera-Leon, J.M. Ramirez, V. Alarcon-Aquino, D. Baker, M. D'Croz-Baron, P. Gomez-Gil, A motor imagery BCI experiment using wavelet analysis and spatial patterns feature extraction, in: 2012 Workshop on Engineering Applications (WEA), IEEE 2012, pp. 1–6.
[8] W. Hsu, C. Lin, H. Hsu, P. Chen, I. Chen, Wavelet-based envelope features with automatic EOG artifact removal: application to single-trial EEG data, Expert Syst. Appl. 39 (3) (2012) 2743–2749.
[9] W. Sweldens, The lifting scheme: a construction of second generation wavelets, SIAM J. Math. Anal. 29 (2) (1998) 511.
[10] J. Asensio-Cubero, J.Q. Gan, R. Palaniappan, Multiresolution analysis over simple graphs for brain–computer interfaces, J. Neural Eng. 10 (4) (2013) 046014.
[11] B.A.S. Hasan, J.Q. Gan, Hangman BCI: an unsupervised adaptive self-paced brain–computer interface for playing games, Comput. Biol. Med. 42 (5) (2012) 598–606.
[12] R. Poli, C. Cinel, A. Matran-Fernandez, F. Sepulveda, A. Stoica, Towards cooperative brain–computer interfaces for space navigation, in: Proceedings of the 2013 International Conference on Intelligent User Interfaces, ACM, 2013, pp. 149–160.
[13] D. Marshall, D. Coyle, S. Wilson, M. Callaghan, Games, gameplay, and BCI: the state of the art, IEEE Trans. Comput. Intell. AI Games 5 (2) (2013) 82–99.
[14] S.K. Narang, A. Ortega, Lifting based wavelet transforms on graphs, Asia-Pacific Signal Inf. Process. Assoc. (2009) 441–444.
[15] E. Martinez-Enriquez, A. Ortega, Lifting transforms on graphs for video coding, in: Data Compression Conference, IEEE, 2011, pp. 73–82.
[16] H. Ramoser, J. Muller-Gerking, G. Pfurtscheller, Optimal spatial filtering of single trial EEG during imagined hand movement, IEEE Trans. Rehab. Eng. 8 (4) (2000) 441–446.
[17] E.A. Mousavi, J.J. Maller, P.B. Fitzgerald, B.J. Lithgow, Wavelet common spatial pattern in asynchronous offline brain–computer interfaces, Biomed. Signal Process. Control 6 (2) (2010) 121–128.
[18] J. Asensio-Cubero, J.Q. Gan, R. Palaniappan, Extracting common spatial patterns based on wavelet lifting for brain computer interface design, in: 2012 4th Computer Science and Electronic Engineering Conference (CEEC), IEEE, 2012, pp. 160–163.
[19] R.A. Fisher, The use of multiple measurements in taxonomic problems, Ann. Eugen. 7 (1936) 179–188.
[20] P. Pudil, J. Novoviov, J. Kittler, Floating search methods in feature selection, Pattern Recognit. Lett. 15 (11) (1994) 1119–1125.
[21] E. Thomas, M. Dyson, M. Clerc, An analysis of performance evaluation for motor-imagery based bci, J. Neural Eng. 10 (3) (2013) 031001.
[22] J. Asensio-Cubero, J.Q. Gan, R. Palaniappan, Wavelet lifting over information-based EEG graphs for motor imagery data classification, Physiol. Comput. Syst., (2014) 3–19.