Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

A P300 based online brain-computer interface system for virtual hand control*

Wei-dong CHEN†[1,2], Jian-hui ZHANG[1,2], Ji-cai ZHANG[1,2], Yi LI[1,2], Yu QI[1,2], Yu SU[1,2], Bian WU[1,3], Shao-min ZHANG[1,3], Jian-hua DAI[1,2], Xiao-xiang ZHENG†‡[1,3], Dong-rong XU[1,4,5]

([1]*Qiushi Academy for Advanced Studies, Zhejiang University, Hangzhou 310027, China*)

([2]*School of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China*)

([3]*Department of Biomedical Engineering, Zhejiang University, Hangzhou 310027, China*)

([4]*MRI Unit, Department of Psychiatry, Columbia University College of Physicians and Surgeons, New York, NY 10032, USA*)

([5]*New York State Psychiatric Institute, New York, NY 10032, USA*)

†E-mail: {chenwd, qaas}@zju.edu.cn

Received Aug. 25, 2009; Revision accepted Jan. 22, 2010; Crosschecked July 1, 2010

Abstract: Brain-computer interface (BCI) is a communication system that can help lock-in patients to interact with the outside environment by translating brain signals into machine commands. The present work provides a design for a virtual reality (VR) based BCI system that allows human participants to control a virtual hand to make gestures by P300 signals, with a positive peak of potential about 300 ms posterior to the onset of target stimulus. In this virtual environment, the participants can obtain a more immersed experience with the BCI system, such as controlling a virtual hand or walking around in the virtual world. Methods of modeling the virtual hand and analyzing the P300 signals are also described in detail. Template matching and support vector machine were used as the P300 classifier and the experiment results showed that both algorithms perform well in the system. After a short time of practice, most participants could learn to control the virtual hand during the online experiment with greater than 70% accuracy.

Key words: Brain-computer interface (BCI), Electroencephalography (EEG), P300, Virtual reality (VR), Template matching, Support vector machine (SVM)

doi:10.1631/jzus.C0910530 **Document code:** A **CLC number:** TP399; R318

1 Introduction

The major goal of brain-computer interface (BCI) is to build an additional pathway between the brain and the external world by interpreting signals of brain activities into machine codes or commands. BCI systems show great promise as a communication tool for patients with severe motor disabilities such as amyotrophic lateral sclerosis (ALS) (Wolpaw *et al*., 2000; Vaughan *et al*., 2003; Lebedev and Nicolelis, 2006; Vaughan and Wolpaw, 2006; Feng *et al*., 2007).

Using electroencephalographic (EEG) signals has gradually been the favorite approach of BCI due to its usability and strong reliability. As one important composition of EEG signal, event-related P300 potential is widely studied. P300 potential is a positive peak about 300 ms after the target stimulus onset in the EEG occurring in response to infrequent significant stimuli (Wolpaw *et al*., 2002). Farwell and Donchin (1988) first employed P300 to construct a BCI system called 'P300 word speller'. The speller presented letters and symbols in a 6×6 matrix and repeatedly flashed each row and column. P300s

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (No. 60873125), the National Institute of Biomedical Imaging and Bioengineering (No. 1R03EB008235-01A1), the Shanghai Commission of Science and Technology (No. 10440710200), and the Fundamental Research Funds for the Central Universities

would be elicited when the target row and column flashed. Zhang *et al*. (2008) developed the asynchronous P300-based BCI and proved the possibility for an asynchronous BCI communication. Moreover, many groups have managed to improve the performance of the P300 word speller, thereby improving both the accuracy and bit-rate to a certain degree (Beverina *et al*., 2003; Serby *et al*., 2005; Sellers *et al*., 2006; Krusienski *et al*., 2008).

Because of the relatively low speed of data transfer and high error rate, currently existing BCI systems are not yet practical or popular. Current trends in BCI systems are to meet the requirements of disabled people and to get the disabled involved in BCI experiments (Piccione *et al*., 2006; Sellers and Donchin, 2006). Nijboer *et al*. (2008) managed to evaluate the efficacy of their BCI communication device for individuals with advanced ALS. The system was reported to work well for those subjects. Hoffmann *et al*. (2008) developed a BCI system for helping the disabled to manipulate electrical appliances. In their experiment, subjects could operate a TV, lamp, door, and window with the P300 potential.

The combination of BCI with external devices rather than just computer screens is believed to extend the use of BCI systems. For example, some groups have reported a higher accuracy with a virtual reality (VR) based BCI system, although it might be unsafe for patients to control certain types of external device (e.g., mechanical hand). Recently, VR has emerged as one solution to overcoming the above-mentioned safety issue. Bayliss *et al*. (2004) integrated VR with the traditional P300 based BCI system and designed a scene that was closer to reality. They constructed a virtual apartment filled with virtual appliances. The participants could control those virtual things via their P300 signals. Piccione *et al*. (2008) implemented a more meaningful VR-based BCI system. In their work, users could control a virtual wheelchair to move from room to room. Finke *et al*. (2009) invented a mind game within the virtual scene based on the P300 potential. A player was required to find a target object via the P300 signal. Zhao *et al*. (2009) introduced an EEG-based driving system in a virtual environment. In addition, some researchers have proposed the use of VR as a medium for improving BCI controls (Friedman *et al*., 2007).

In 2007, we developed a P300-based Chinese typewriter (Su *et al*., 2008). Disabled persons such as ALS patients can use this system to print sentences in Chinese with P300 signals. In 2008, we further developed a BCI messenger system based on the P300 typewriter, by which the disabled can send typed messages to cell phones (Li *et al*., 2009). Functions of typing non-Chinese characters including symbols were added to this typewriter and we also optimized many parameters critical to the success of the experiments.

We herein present a P300-based online BCI system for controlling a virtual hand. We developed this system for participants to receive necessary training on BCI in advance so that their controlling skill could be improved, with the consideration that naive participants may find too much difficulty or it could be unsafe to control a real hand or an actual artificial limb without any experience on a BCI system. Meanwhile, VR is perfectly safe for these BCI systems. The P300-based BCI system described here allows a participant to perform a number of common hand gestures. This VR based BCI system aims to prototype practical solutions for various BCI applications.

We therefore demonstrate the possibility of interacting with the real world through BCI systems as long as a similar system can be implemented in a virtual environment, which is almost always feasible.

2 System model

2.1 System and design

A conventional P300-based BCI system usually contains three major modules: data acquisition, signal processing, and an interface for the participant. The module of data acquisition records EEG data from the brain and passes the data to the module of signal processing. This second module extracts features from raw EEG data, recognizes patterns (in our example, patterns corresponding to specific hand gestures) from the features, and then converts the recognized patterns into feedback signals. Finally, the third module displays stimulus scene and feedback information to the participant.

This virtual hand controlling system basically follows this design convention (Fig. 1), but contains four modules for one monitor and a data management

module was added. The module of data acquisition employed a function provided by Acquire software of Neroscan Corporation. The module of signal processing was implemented using Microsoft Foundation Classes (MFC), thereby bridging the first and the third modules. This system does not require high-end computers and can use a laptop as a server for processing the brain signals. In the user interface module, we developed in-house a program to create eliciting solutions for the P300 experiment. This eliciting function contains all the gestures that are supported by our virtual hand and a participant can choose the desired target via his/her P300 potential. The feedback module, which contains a 3D virtual hand, would display the selected virtual hand gesture. This module performs a gesture realistically after every selection of the P300 potential. In our experiment, the feedback module has been embedded into the kernel of the stimulus program, so that the participant may see the stimulus and the feedback simultaneously. Lastly, a monitor module helps the operator to achieve a better control of the system. We will elaborate the technical details in the following subsections.

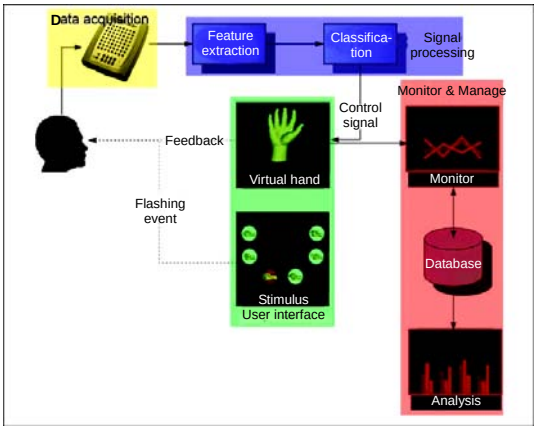

Fig. 1 Our brain-computer interface system containing four major modules: data acquisition, signal processing, user interface, and monitor & manage

2.2 Task and procedure

Compared to other EEG based BCI systems (e.g., BCI based on motor imagination), a P300-based system requires less or even no prior training. One main reason for this is that P300 potential is easy to elicit and the waveform can be retained for quite a long period of time. Therefore, our system requires a pre-flight stage of only 20 min (10 runs and 200 P300 waves averaged). Although only a short pre-flight stage is called for, this procedure is still very important, for the classification algorithm of feature extraction in the module of data acquisition as well as for the participants to familiarize themselves with the system. This classification algorithm calculates parameters used at the next stage when a participant may gain some initial BCI experience on the system and may subsequently adapt to the BCI experiment.

Four human participants (aged 21–25, one female) attended the experiments, and they had all attended the experiments of our previous project (Li *et al.*, 2009). Written consent was obtained. The entire experiment consisted of two stages: a pre-flight stage and a feedback stage. The difference between these two stages was that no feedback was provided in the pre-flight stage. Initially, a participant was told to gaze at one of the gesture images (stimuli) around a virtual hand at the center of the computer screen (Fig. 2). When the stimulus started flashing, the participant was instructed to count the number of the desired stimuli flashing. After seven flashes for each gesture, the participant would see the virtual hand performing a desired action if the corresponding signal processing algorithm worked well. Between each two selections, the participant would be allowed a relatively long break so that the participant could get ready for the next gesture and then repeat the above procedure.

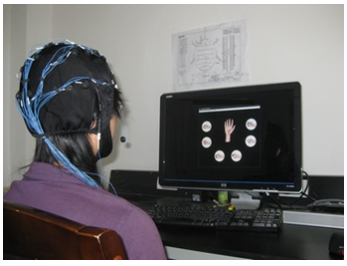

Fig. 2 A participant sitting in front of the monitor during the online experiment

All the participants were asked to take two sessions of experiment in two consecutive weeks. In the first session, the participants finished the pre-flight stage training, which lasted about 20 min. About 10 runs were included in one session and each run contained about 8–12 selections. Thus, we collected for each participant a total of about 200 trials of P300

data in the feedback stage.

All the experiments were undertaken with inter-stimulus intervals (ISI, onset-to-onset time interval for stimulus) of 200 ms, as was found from our previous research (Li *et al*., 2009) that an ISI of 200 ms was preferred to either longer or shorter ISIs in that, longer intervals tend to cause anxiety and impatience in the participants whereas shorter intervals require higher concentration. Consequently, one run would need 3–5 min and an entire session including 20 runs would take about 75 min.

2.3 Data acquisition

EEG was recorded using Quick-Cap (Compumedics Neuroscan) embedded with 37 electrodes distributed over the entire scalp. Fourteen channels were used in this experiment (FZ, FC3, FCZ, FC4, T3, C3, CZ, C4, T4, CP3, CPZ, CP4, PZ, and OZ), all referenced to the nose and grounded to the forehead. The acquired data were digitized at a sample rate of 250 Hz, band pass filtered at 0.1–40 Hz, and then stored. Vertical electrooculogram (EOG) was recorded for ocular artifact removal. All the EEG data and event code of the stimuli were recorded and sent to the signal processing module during the online experiment.

The experimental controlling and information recording were finished by the experiment monitor module. It recorded some vital details (e.g., flashing events, feedback sequence, and time cost) of the experiment and also helped operators to better control the whole procedure. Some data management and statistical analysis functions were also integrated into this program, and we could obtain additional assistance during an offline data processing procedure.

2.4 Virtual hand modeling

The human hand is a complex biological organ; it can be considered a mechanical machine. To construct a virtual hand in 3D environment for various immersive or non-immersive applications, the human hand's biomechanical features must be fully considered. In modeling a virtual hand, three key aspects must be considered: the physical modeling of the hand, the constraints of hand motion, and the real-time rendering of hand gestures.

Different from medical researchers and biomechanical scientists who might be more interested in sophisticated methods and measurements to mimic hand motions and model hand structures, we as computer scientists and engineering professionals consider that the ability of interaction and the speed of rendering are more important aspects (Yi *et al*., 2005). To drive the virtual hand with EEG signals on-the-fly, we accepted a trade-off between the visual quality of appearance of the hand model and the real-time interactivity.

A human hand comprises 27 bones and 29 degrees of freedom (DoFs). Hand motions result from complex combinations of the movements of these bones with various joints, each having differing DoFs (Sturman, 1992). A virtual hand is ideally modeled with all the 27 bones, joints and all the 29 DoFs. For the above-mentioned trade-off, however, we employed a simplified hand model that is visually pleasant and acceptable with a reduced number of DoFs.

Our model considered the virtual hand as a rigid body, with no physical deformation but merely a set of several simple geometries, at a size of thousands of polygons in predefined order. We divided the virtual hand into 15 parts, of which 12 parts were for finger phalanxes, two for thumb phalanxes, and one large piece for the palm. Therefore, this virtual hand has 14 joints with about 24 DoFs: PIP (proximal interphalangeal) joint and DIP (distal interphalangeal) joint for four fingers with one degree each; MCP (metacarpophalangeal) joint for four fingers with two degrees each, except for the middle finger which has only one; IP (interphalangeal) joint of the thumb with one degree and MCP of the thumb with two degrees; the wrist with the last six degrees. The whole hand is organized as a virtual scene graph, with the root node representing the wrist. Each part of the virtual hand model has its own coordinate system and is attached to its parent node, which has a different coordinate system. For example, the distal phalanx of the middle finger has its own coordinate system rooted at its DIP joint, while this coordinate system is attached to the coordinate system of its middle phalanx rooted at its PIP joint. The scene graph of the hand structure is a tree-like structure that roots from the wrist joint of the hand. Each joint plays a role as a node of the tree, and the phalanxes and bones as leaf nodes.

Even though a human hand is highly articulated and fingers can move naturally without external interaction, the hand cannot generate arbitrarily random

gestures. The anatomical structure highly constrains the possible hand gestures. Natural movements of hand parts at the joints are therefore spatially limited. An evident constraint is in a single finger flexion, which might cause flexion of the adjacent fingers, and this finger's extension is hindered by the flexion of others. There are three types of hand constraint (Lin *et al*., 2000):

1. The constraints limited by the finger motion as a result of hand anatomy, called static constraints (i.e., joint motion ranges).

2. The constraints imposed on joints during motion, or dynamic constraints (i.e., the intra- and inter-finger constraints).

3. The constraints applied in performing natural motions (i.e., curling all fingers at the same time to make a fist).

When the motion of a virtual hand is modeled, applying hand constraints can significantly improve the computational efficiency and provide visually realistic animation of hand motion. With the reduced number of joints and freedoms in our hand model, the constraints applied from studies on fingers and thumb motions are calculated as follows:

1. The joint angle static limits of MCP, PIP, and DIP joints for the four fingers as well as the MCP and IP joints for thumb are indicated in Tables 1 and 2.

Table 1 Finger joint motion ranges (degree)

Joint	Motion	Index	Middle	Ring	Pinky
DIP	Flexion	73	80	75	78
	Extension	11	11	11	11
PIP	Flexion	101	103	105	103
	Extension	12	12	12	7
MCP	Flexion	83	90	88	90
	Extension	−22	−22	−23	−34

DIP: distal interphalangeal; PIP: proximal interphalangeal; MCP: metacarpophalangeal

Table 2 Thumb joint motion ranges

Joint	Motion	Thumb (°)
IP	Flexion-extension	100±9
	Abduction-adduction	7±10
	Rotation	8±9
MCP	Flexion-extension	45±16
	Abduction-adduction	9±3
	Rotation	12±4

IP: interphalangeal; MCP: metacarpophalangeal

2. The 5DT® data glove we used does not have DIP sensors for the four fingers. The DIP and PIP joint flexions are not independent.

$$\theta^{\mathrm{F}}_{\mathrm{DIP}} = \frac{2}{3}\theta^{\mathrm{F}}_{\mathrm{PIP}}. \tag{1}$$

3. The MCP joint of the middle finger has no abduction or adduction.

$$\theta^{\mathrm{A}}_{\mathrm{MCP(M)}} = 0. \tag{2}$$

We also applied the inter-finger dynamic constraint, which states that the joint angle limits of the MCP joints depend on those of the neighboring fingers. That is, how one finger moves at its MCP joint can affect its neighboring fingers. For example, when the flexion increases at the MCP joint of the ring finger, both the middle and little fingers will follow the ring finger. In this case, the ring finger is active while the middle and little fingers are passive. The state of the ring finger at the MCP joint depends on the active finger's flexion angle as well as the passive finger proximity to the active one. That is, the middle and little fingers both have a strong influence on the ring finger while the index has moderate and the thumb the least influence. Since there is a maximum difference in flexion angles between active and passive fingers, the influence of passive fingers on active fingers can be summarized as follows (Lee and Kunii, 1995):

$$\alpha_{\mathrm{d\,max}}(\theta^{\mathrm{F}}_{\mathrm{MCP(I)}}) = \min\left(\theta^{\mathrm{F}}_{\mathrm{MCP(M)}} + 25, \alpha_{\mathrm{s\,max}}(\theta^{\mathrm{F}}_{\mathrm{MCP(I)}})\right), \tag{3}$$

$$\alpha_{\mathrm{d\,min}}(\theta^{\mathrm{F}}_{\mathrm{MCP(I)}}) = \max\left(\theta^{\mathrm{F}}_{\mathrm{MCP(M)}} - 54, \alpha_{\mathrm{s\,min}}(\theta^{\mathrm{F}}_{\mathrm{MCP(I)}})\right), \tag{4}$$

$$\alpha_{\mathrm{d\,max}}(\theta^{\mathrm{F}}_{\mathrm{MCP(M)}}) = \min\left(\theta^{\mathrm{F}}_{\mathrm{MCP(I)}} + 54, \theta^{\mathrm{F}}_{\mathrm{MCP(R)}} + 20, \alpha_{\mathrm{s\,max}}(\theta^{\mathrm{F}}_{\mathrm{MCP(M)}})\right), \tag{5}$$

$$\alpha_{\mathrm{d\,min}}(\theta^{\mathrm{F}}_{\mathrm{MCP(M)}}) = \min\left(\theta^{\mathrm{F}}_{\mathrm{MCP(I)}} - 25, \theta^{\mathrm{F}}_{\mathrm{MCP(R)}} - 45, \alpha_{\mathrm{s\,min}}(\theta^{\mathrm{F}}_{\mathrm{MCP(M)}})\right), \tag{6}$$

$$\alpha_{\mathrm{d\,max}}(\theta^{\mathrm{F}}_{\mathrm{MCP(R)}}) = \min\left(\theta^{\mathrm{F}}_{\mathrm{MCP(M)}} + 45, \theta^{\mathrm{F}}_{\mathrm{MCP(L)}} + 48, \alpha_{\mathrm{s\,max}}(\theta^{\mathrm{F}}_{\mathrm{MCP(R)}})\right), \tag{7}$$

$$\alpha_{\mathrm{d\,min}}(\theta^{\mathrm{F}}_{\mathrm{MCP(R)}}) = \min\left(\theta^{\mathrm{F}}_{\mathrm{MCP(M)}} - 20, \theta^{\mathrm{F}}_{\mathrm{MCP(L)}} - 44, \alpha_{\mathrm{s\,max}}(\theta^{\mathrm{F}}_{\mathrm{MCP(R)}})\right), \tag{8}$$

$$\alpha_{\mathrm{d\,max}}(\theta^{\mathrm{F}}_{\mathrm{MCP(L)}}) = \min\left(\theta^{\mathrm{F}}_{\mathrm{MCP(R)}} + 44, \alpha_{\mathrm{s\,max}}(\theta^{\mathrm{F}}_{\mathrm{MCP(L)}})\right), \tag{9}$$

$$\alpha_{\mathrm{d\,min}}(\theta^{\mathrm{F}}_{\mathrm{MCP(L)}}) = \max\left(\theta^{\mathrm{F}}_{\mathrm{MCP(R)}} - 48, \alpha_{\mathrm{s\,min}}(\theta^{\mathrm{F}}_{\mathrm{MCP(L)}})\right), \tag{10}$$

where $\alpha_{\mathrm{d}\max}$ and $\alpha_{\mathrm{d}\min}$ are the maximum and minimum dynamic angles at each joint respectively, subscripts I, M, R, and L represent the index, middle, ring, and little fingers respectively, and $\alpha_{\mathrm{s}\max}$ and $\alpha_{\mathrm{s}\min}$ are static joint angles as listed in Table 1.

Based on this hand model and its motion constraints, we are then able to generate hand motions. Investigators have found that human hands can manage two classes of hand motion: prehensile and non-prehensile. Prehensile motions are typically gripping or pinching an object between the digits and the palm, for grasping an object or using a tool. Typical non-prehensile motions include pushing, lifting, and tapping.

We defined seven hand gestures in our experiment for elementary tasks (Fig. 3). The gesture of hand completely open with all available DoFs at zero positions was defined as the reference gesture, and the other six gestures were: hand half open for ball grasping, full close of the hand to a fist, hand opposition, hand grip, hand gesture for number two, and the OK sign. The gestures are generated based on data captured using a 5DT® data glove.

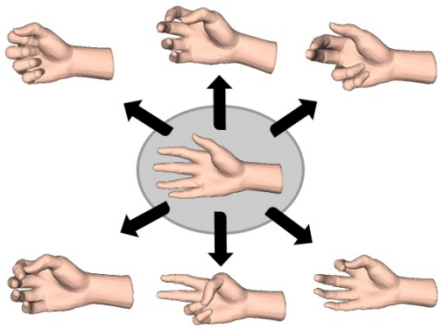

Fig. 3 Virtual hand gestures supported by the system
The six gestures from top down and left to right are: hand full close to a fist, half open for ball grasping, opposition, grip, gesture for number two, and the OK sign

2.5 Signal processing

The common signal processing procedure of a BCI system contains two stages: feature extraction and classification. Different classification methods were employed in our online experiment and offline data analysis. In this subsection, we first introduce the feature extraction method, and then describe the online classification algorithm and offline data analysis method in detail.

EEG signal is recorded by the electrodes far from the signal source, so its signal-to-noise ratio (SNR) is low. Although the signals may be characterized by distinct prominent features such as P300, distinguishing the differing signals remains a challenging job. Preprocessing of the signals is therefore necessary to improve SNR by reducing artifacts and correcting the signal drifts. Four steps were involved in our feature extraction procedure: artifact reduction, drift correction, signal filtering, and averaging.

Ocular movements, especially blinks, have a large negative influence on the entire EEG waveform. In the P300 experiments, the classification algorithm could create significant errors if the participant blinked while counting the target. Unfortunately, the participants cannot prevent blinking during the whole period of stimulus flashing. Therefore, the impact of EOG on the EEG wave has to be isolated and removed.

The algorithm for eliminating the EOG we used employs least squares regression (Croft and Barry, 2000). The proportion of EOG that is present in the *k*th EEG channel is given by

$$B_k=\frac{\sum(\mathrm{EOG}_i-\overline{\mathrm{EOG}_i})(\mathrm{EEG}_i-\overline{\mathrm{EEG}_i})}{\sum(\mathrm{EOG}_i-\overline{\mathrm{EOG}_i})^2}, \quad (11)$$

where EOG_i and EEG_i are the amplitudes of EOG and the *k*th EEG at time point *i* respectively, and $\overline{\mathrm{EOG}_i}$ and $\overline{\mathrm{EEG}_i}$ are the mean values of EOG and EEG data from all channels at time point *i* respectively. Then we calculated the corrected amplitude of the *k*th EEG by

$$\mathrm{EEG}_i'=\mathrm{EEG}_i-(\overline{\mathrm{EOG}_i}-B_k)-(\overline{\mathrm{EOG}_i}-(\overline{\mathrm{EEG}_i}\times B_k)). \quad (12)$$

Wave drifting is another basic problem in the experiment. The EEG wave we obtained from the amplifier always drifted either upward or downward. Some specific waveform character was concealed and became more difficult to recognize. So a drifting correction procedure was employed at this stage to diminish this negative influence. For every *m* points in the EEG wave, the first *p* continuous points were selected and the average value was calculated. In our case, m=50 and p=10.

$$\mathrm{AVG}_i = \frac{1}{p}\sum_{j=im+1}^{im+p} \mathrm{EEG}_j. \quad (13)$$

With the calculated point array AVG, a line that approximated the drifting path could be simulated. Then the point value of the line (Δ) was subtracted from the original EEG wave on every point to obtain the corrected EEG signal.

$$\mathrm{EEG}' = \mathrm{EEG} - \Delta. \quad (14)$$

Then, we filtered and down-sampled the signals. We filtered the corrected EEG data using a band-pass filter (0.1–15 Hz) to exclude components irrelevant to P300 potential. In practice, every three sampling points were replaced by their average value to reduce the resolution of the EEG signals.

Because single trial P300 was generally immersed into background EEG, averaging signal over trials turned out a reasonable method for enhancing P300. As the background EEG signals are generally random, whereas P300 signals are not, averaging EEG signals at the eliciting time can weaken the background EEG signals' influence. Subsequently, we segmented the continuous EEG data into epochs, each containing 620 ms of signals, from a time point at 100 ms after the stimulus flashing to 720 ms afterwards. Then, epochs of seven trials were averaged to obtain an enhanced epoch for each stimulus. The P300 waveform would become obvious after this step. Fig. 4 shows P300 and non-P300 waveforms of a participant, which were averaged over seven trials at FCZ.

After the above feature extraction procedure, the P300 waveform stood out. What was still needed was an effective classifier for higher accuracy and faster processing in the online system. The algorithm of template matching was adopted in our online experiment. Template matching is a simple algorithm suitable for waveform recognition and therefore appropriate for detecting P300 signals. Moreover, this method is easy to implement and very fast in classification, important in our online system requiring real-time processing.

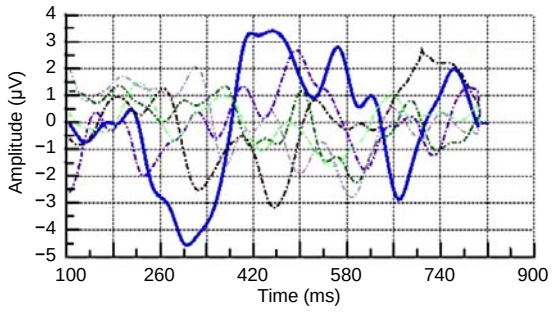

Fig. 4 Typical P300 waveform of one participant after averaging over seven trials at FCZ
The solid line is the target P300 wave; others are non-target waves

In the online processing step, epochs corresponding to every stimulus were calculated and stored in a matrix. If every epoch had m points, and n channels were employed, the matrix turned to be an $m\times n$ vector. After averaging the initial 10 epochs via the pre-flight stage, we obtained the P300 template for the subject. Then, the similarity of two matrixes $\boldsymbol{A}$ and $\boldsymbol{B}$ could be calculated using Eq. (15), where $\boldsymbol{A}_i$ and $\boldsymbol{B}_i$ are the ith column vectors of matrixes $\boldsymbol{A}$ and $\boldsymbol{B}$, respectively. If a wave resembles P300 template wave, its value of similarity to the P300 template is high.

$$S(\boldsymbol{A}, \boldsymbol{B}) = \sum_{i=1}^{n} S(\overline{\boldsymbol{A}_i}, \overline{\boldsymbol{B}_i}). \quad (15)$$

Calculating the similarity of two matrixes could be viewed as calculating the similarity sum of two groups of channels. A method of correlation coefficients as defined in Eq. (16) was used to calculate the similarity of two vectors $\boldsymbol{X}$ and $\boldsymbol{Y}$:

$$S = \frac{\sum x_i y_i}{\sqrt{\sum x_i^2 \sum y_i^2}}. \quad (16)$$

We therefore obtained n similarity values corresponding to the n stimulus events. The epoch that had the largest similarity would be selected as the target. But if the difference between the largest and second largest similarities was small, no target would be chosen. The target value, once selected, would be sent to the feedback module, and the corresponding gesture would be performed by the virtual hand.

2.6 Offline analysis

In the offline data processing stage, a machine learning method called support vector machine (SVM) was applied to classify the epoch data. In pursuit of real-time interaction, we chose template matching

(TM) as the online classification method. Nevertheless, SVM is widely considered to be superior to TM in terms of classification accuracy.

Generally speaking, the idea of support vector classifier is to maximize the margin between the training patterns and the decision boundary (Theodoridis and Koutroumbas, 2006). As shown in Fig. 5, classifier $g_1(\boldsymbol{x})$ is better than classifier $g_2(\boldsymbol{x})$, because the geometric margin of $g_1(\boldsymbol{x})$ is larger than that of $g_2(\boldsymbol{x})$. $g_1(\boldsymbol{x})$ may therefore perform better in predicting future data.

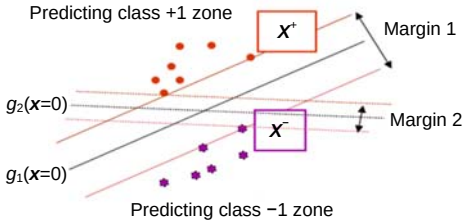

Fig. 5 An example of support vector machine decision, where classifier $g_1(x)$ is better than classifier $g_2(x)$ owing to a larger margin

Cortes and Vapnik (1995) suggested a modified maximum margin method which allowed for mislabeled examples (Boser *et al*., 1992). The method introduces slack variables ξ_i, which measure the degree of misclassification of data. If there is no hyperplane that can split the '+1' or '−1' examples, the method will choose a hyperplane that splits the examples as cleanly as possible, while still maximizing the distance to the nearest cleanly split examples. This is called the soft margin method. By penalizing non-zero ξ_i, soft margin makes a trade-off between margin and error penalty. This kind of support vector classifier takes the following form:

Given the training data of two classes which are labeled as −1 or +1,

$$\{\boldsymbol{x}_i, y_i\},\ i=1, 2,..., N,\ y_i \in \{-1,\ 1\},\ \boldsymbol{x}_i \in \mathbb{R}^m, \quad (17)$$

finding the soft margin classifier becomes a quadratic programming (QP) optimization problem:

$$\begin{aligned} &\min\left(\frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N}\xi_i\right)\\ &\text{s.t.}\quad y_i(<\boldsymbol{w}, \boldsymbol{x}_i> + w_0) \geq 1-\xi_i \\ &\qquad \xi_i \geq 0,\ i=1,2,...,N. \end{aligned} \quad (18)$$

Solving the QP problem, we obtain the classifier:

$$g(\boldsymbol{x}) = \boldsymbol{w}\cdot\boldsymbol{x} + w_0 > (<)\ 0. \quad (19)$$

Moreover, the above QP problem has a duality form of

$$\begin{aligned} &\max_{\lambda}\left(\sum_{i=1}^{N}\lambda_i - \frac{1}{2}\sum_{i,j}\lambda_i\lambda_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)\right)\\ &\text{s.t.}\qquad \sum_{i=1}^{N}\lambda_i y_i = 0,\quad 0 \leq \lambda \leq C. \end{aligned} \quad (20)$$

And the classifier in this condition becomes

$$g(\boldsymbol{x}) = \boldsymbol{w}\cdot\boldsymbol{x} + w_0 = \sum_{i=1}^{N_s}\lambda_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + w_0 > (<)\ 0, \quad (21)$$

where N_s is the number of non-zero support vectors and $K(\boldsymbol{x}_i,\ \boldsymbol{x}_j)$ is a kernel function, which maps the original data to a transformed feature space. The transformation may be non-linear and the transformed space can be high dimensional; thus, although the classifier is a hyperplane in the high-dimensional feature space, it may be non-linear in the original input space.

In this study, we chose the dot product as the kernel function and implemented a soft margin linear classifier. In the prediction phase, with the classifier

$$g(\boldsymbol{x}) = \boldsymbol{w}\cdot\boldsymbol{x} + w_0 = \sum_{i=1}^{N_s}\lambda_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + w_0, \quad (22)$$

we assign $\boldsymbol{x}$ to the '+1' (or '−1') class if $g(\boldsymbol{x})>0$ (or <0). Since $g(\boldsymbol{x})$ is a distance measured from $\boldsymbol{x}$ to the hyperplane, we may take advantage of this using the value instead of just the sign of $g(\boldsymbol{x})$. In the P300 classification task, we have to pick one averaged P300 epoch out of N epochs while the others are non-target epochs. Here N is the number of stimuli in a typical oddball experiment. If we label the target P300 epochs as '+1' class and the non-target epochs as '−1' class, we can conclude that the epoch with the maximum $g(\boldsymbol{x})$ value in all N epochs contains the target P300. This method performed well in our experiments and further analysis about this algorithm will be made in the near future studies.

3 Results

Compared to the P300 experiments conducted before, experiments in this study of the virtual hand control are somewhat different, and some differences may have crucial effect on the final result. First, virtual reality was employed as the feedback medium. Feedback from VR provides realistic visual effects, but nevertheless may distract a participant from the stimulus. Furthermore, feedback from VR would be less intuitive, and it could take more time for the participant to obtain the results. Second, we also tried a reduced number of trials for average. In this study we set the trial number to be 7 (in contrast to 10 in former research), resulting in decreased accuracy in the final classification, but an improved rate of data transfer.

All participants, except participant B, finished the online experiment (TM method) with an accuracy higher than 70% (Fig. 6). To testify the effectiveness of the SVM algorithm, a semi-online procedure was designed in which the training data were used for training the SVM classifier while the other data for offline checking. We thereby obtained the average accuracy for every participant. The average accuracy of all participants using the SVM classifier was 67.96%, lower than the TM method used online (72.38%). The SVM classifier's average accuracy, however, reached 73.91% while the TM method achieved only 65.94% for participants A, B, and C. On the whole, our offline method appeared to be a better solution for the system provided that the performance for some individuals (e.g., participant D) could be improved.

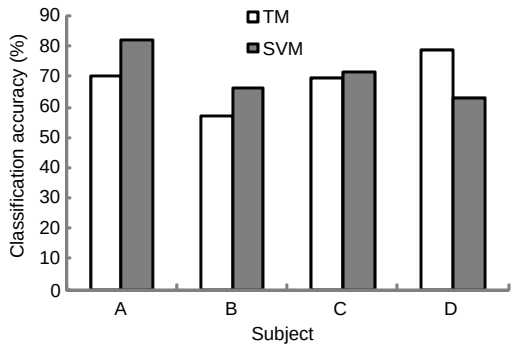

Fig. 6 Comparison of classification accuracy for the online experiment (template matching, TM) and offline analysis (support vector machine, SVM)

To compare the performances of the online and semi-online algorithms, some more experiments were carried out. We exchanged the methods used in the online and semi-online stages for several trials. The results showed that the method of working online always outperformed that of working semi-online. The reason may be that the online experiment allowed a participant to adapt himself/herself to the algorithm after each selection. So the algorithm would always be better when it is working online than semi-online. This may be due to the mutual adaption between the participants and the algorithm. As a matter of fact, one of the participants performed well in the P300 tasks even when the data processing algorithm had some defects. We are certain that humans have strong ability in learning rules and controlling the outside systems such as BCIs. More work is required to examine this perspective.

Compared with our previous studies, we obtained a little lower accuracy for our virtual hand system. One main reason might be the reduction of the number of trials for each selection. Averaging was an important procedure for enhancing the P300 wave, but largely reduced the bit-rate of the BCI system. Although the system accuracy became lower than that of the previous work, where 10-trial averaging was used, the transfer rate achieved a 2.62% increase.

Reviewing the online classification results, we found that the selection accuracy became much higher when the last selection was correct. That is to say, the selection was apt to be correct when a correct choice had been made. In fact, the participants' accuracy was on average 11% higher when they had already made a correct choice. This pointed to the importance of human adaption during the P300 experiment. Positive feedback reinforced the learning procedure, while negative feedback suggested the participant to adapt. The positive feedback seems more important for participants to maintain better performance states as it may relax the participant, while a negative result may increase tension in the participant. Compared to other types of interaction method, mental factors such as mood and psychology appear to play a more important role in BCI.

Moreover, we tried to analyze the contribution of each individual channel to P300 classification. Fig. 7 shows the Encephalo topography for four participants. The value of each point indicates the channel's simi-

larity to the template, i.e., the contribution of each channel. For all the participants, the regions of channels C3 and CP3 showed an obviously better effect. However, the traditional channel group used for the P300 experiment (FCZ, CZ, CPZ, and PZ) seems less contributive in our P300 detection method. This was particularly the case for the participants A, B, and C, whose Encephalo topographies were very similar. This similarity can be very important for finding a general channel setup for a P300 experiment. We also found that reducing the channel number shortened the preparation time for the experiment.

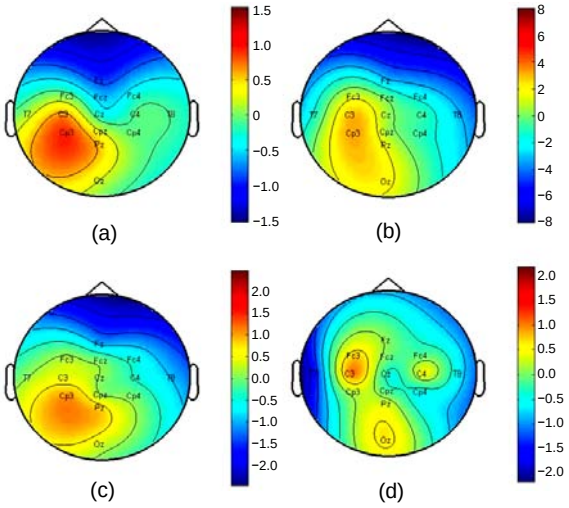

Fig. 7 Encephalo topography of online experiment results with four participants A (a), B (b), C (c), and D (d)
The value of each point indicates the channel's similarity to the template, i.e., the contribution of each channel

4 Conclusions

With brain-computer interface (BCI), disabled persons such as amyotrophic lateral sclerosis (ALS) patients could better care for themselves. In this paper, we described a P300-based online virtual hand controlling system and the signal processing method. All participants of the experiment could control the virtual hand correctly after a short time of practice. We also demonstrated that, the system's transfer rate increased when the trial number was reduced, although the accuracy of the system was not stable and performance was not very good for some participants. To achieve a more practical and faster BCI system, more work needs to be done to improve our algorithm. A channel analysis was also conducted, showing that some channels were more contributive than the others. Virtual reality can provide the disabled with a very good training environment for BCI use. The combination of VR and BCI seems very promising for BCI systems in terms of its usability and practicability.

References

Bayliss, J.D., Inverso, S.A., Tentler, A., 2004. Changing the P300 brain computer interface. *Cyberpsych. & Behav.*, **7**(6):694-704. [doi:10.1089/cpb.2004.7.694]

Beverina, F., Palmas, G., Silvoni, S., Piccione, F., 2003. User adaptive BCIs SSVEP and P300 based interfaces. *PsychNol. J.*, **1**(4):331-354.

Boser, B.E., Guyon, I.M., Vapnik, V., 1992. A Training Algorithm for Optimal Margin Classifiers. Proc. 5th Annual Workshop on Computational Learning Theory, p.144-152. [doi:10.1145/130385.130401]

Cortes, C., Vapnik, U., 1995. Support-vector networks. *Mach. Learn.*, **20**(3):273-297. [doi:10.1023/A:1022627411411]

Croft, R.J., Barry, R.J., 2000. Removal of ocular artifact from the EEG: a review. *Neurophys. Clin.*, **30**(1):5-19. [doi:10.1016/S0987-7053(00)00055-1]

Farwell, L.A., Donchin, E., 1988. Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephal. Clin. Neurophys.*, **70**(6):510-523. [doi:10.1016/0013-4694(88)90149-6]

Feng, Z.Y., Chen, W.D., Ye, X.S., Zhang, S.M., Zheng, X.J., Wang, P., Jiang, J., Jin, L., Xu, Z.J., Liu, C.Q., *et al*., 2007. A remote control training system for rat navigation in complicated environment. *J. Zhejiang Univ. Sci. A*, **8**(2):323-330. [doi:10.1631/jzus.2007.A0323]

Finke, A., Lenhardt, A., Ritter, H., 2009. The MindGame: a P300-based brain computer interface game. *Neur. Networks*, **22**(9):1329-1333. [doi:10.1016/j.neunet.2009.07.003]

Friedman, D., Leeb, R., Guger, C., Steed, A., Pfurtscheller, G., 2007. Navigating virtual reality by thought: what is it like? *Pres. Teleoper. Virt. Environ.*, **16**(1):100-110. [doi:10.1162/pres.16.1.100]

Hoffmann, U., Vesin, J.M., Ebrahimi, T., Diserens, K., 2008. An efficient P300-based brain-computer interface for disabled subjects. *J. Neuros. Methods*, **167**(1):115-125. [doi:10.1016/j.jneumeth.2007.03.005]

Krusienski, D.J., Sellers, E.W., McFarland, D.J., Vaughan, T.M., Wolpaw, J.R., 2008. Toward enhanced P300 speller performance. *J. Neurosci. Methods*, **167**(1):15-21. [doi:10.1016/j.jneumeth.2007.07.017]

Lebedev, M.A., Nicolelis, M.A., 2006. Brain-machine interfaces: past, present and future. *Trends Neurosci.*, **29**(9):536-546. [doi:10.1016/j.tins.2006.07.004]

Lee, J., Kunii, T.L., 1995. Model-based analysis of hand posture. *IEEE Comput. Graph. Appl.*, **15**(5):77-86. [doi:10.1109/38.403831]

Li, Y., Zhang, J.H., Su, Y., Chen, W.D., Qi, Y., Zhang, J.C., Zheng, X.X., 2009. P300 Based BCI Messenger.

IEEE/ICME Int. Conf. on Complex Medical Engineering, p.1-5. [doi:10.1109/ICCME.2009.4906636]

Lin, J., Wu, Y., Huang, T.S., 2000. Modeling the Constraints of Human Hand Motion. Proc. Workshop on Human Motion, p.121-126. [doi:10.1109/HUMO.2000.897381]

Nijboer, F., Sellers, E.W., Mellinger, J., Jordan, M.A., Matuz, T., Furdea, A., Halder, S., Mochty, U., Krusienski, D.J., Vaughan, T.M., *et al*., 2008. A P300-based brain-computer interface for people with amyotrophic lateral sclerosis. *Clin. Neurophys.*, **119**(8):1909-1916. [doi:10.1016/j.clinph.2008.03.034]

Piccione, F., Giorgi, F., Tonin, P., Priftis, K., Giove, S., Silvoni, S., Palmas, G., Beverina, F., 2006. A P300-based brain computer interface reliability and performance in healthy and paralysed participants. *Clin. Neurophys.*, **117**(3):531-547. [doi:10.1016/j.clinph.2005.07.024]

Piccione, F., Priftis, K., Tonin, P., 2008. Task and stimulation paradigm effects in a P300 brain computer interface exploitable in a virtual environment: a pilot study. *PsychNol. J.*, **6**(1):99-108.

Sellers, E.W., Donchin, E., 2006. A P300-based brain-computer interface: initial tests by ALS patients. *J. Neurosci. Methods*, **117**:538-548. [doi:10.1016/j.clinph.2005.06.027]

Sellers, E.W., Krusienski, D.J., McFarland, D.J., Vaughan, T.M., Wolpaw, J.R., 2006. A P300 event-related potential brain-computer interface (BCI): the effects of matrix size and inter stimulus interval on performance. *Biol. Psychol.*, **73**(3):242-252. [doi:10.1016/j.biopsycho.2006.04.007]

Serby, H., Yom-Tov, E., Inbar, G.F., 2005. An improved P300-based brain-computer interface. *IEEE Trans. Neur. Syst. Rehabil. Eng.*, **13**(1):89-98. [doi:10.1109/TNSRE.2004.841878]

Sturman, D.J., 1992. Whole-Hand Input. PhD Thesis, Massachusetts Institute of Technology, USA.

Su, Y., Wu, B., Chen, W.D., 2008. P300-based brain computer interface: prototype of a Chinese speller. *J. Comput. Inform. Syst.*, **4**(4):1515-1522.

Theodoridis, S., Koutroumbas, K., 2006. Pattern Recognition. Elsevier/Academic Press, Boston.

Vaughan, T.M., Wolpaw, J.R., 2006. The third international meeting on brain-computer interface technology: making a difference. *IEEE Trans. Neur. Syst. Rehabil. Eng.*, **14**(2):126-127. [doi:10.1109/TNSRE.2006.875649]

Vaughan, T.M., Heetderks, W.J., Trejo, L.J., Rymer, W.Z., Weinrich, M., Moore, M.M., Kübler, A., Dobkin, B.H., Birbaumer, N., Donchin, E., *et al*., 2003. Brain-computer interface technology: a review of the second international meeting. *IEEE Trans. Neur. Syst. Rehabil. Eng.*, **11**(2):94-109. [doi:10.1109/TNSRE.2003.814799]

Wolpaw, J.R., Birbaumer, N., Heetderks, W.J., McFarland, D.J., Peckham, P.H., Schalk, G., Donchin, E., Quatrano, L.A., Robinson, C.J., Vaughan, T.M., 2000. Brain-computer interface technology: a review of the first international meeting. *IEEE Trans. Rehabil. Eng.*, **8**(2):164-173. [doi:10.1109/TRE.2000.847807]

Wolpaw, J.R., Birbaumer, N., McFarland, D.J., Pfurtscheller, G., Vaughan, T.M., 2002. Brain-computer interfaces for communication and control. *Clin. Neurophys.*, **113**(6):767-791. [doi:10.1016/S1388-2457(02)00057-3]

Yi, B., Harris, F.C.Jr., Wang, L., Yan, Y., 2005. Real-time natural hand gestures. *Comput. Sci. Eng.*, **7**(3):92-96. [doi:10.1109/MCSE.2005.58]

Zhang, H.H., Guan, C.T., Wang, C.C., 2008. Asynchronous P300-based brain-computer interfaces: a computational approach with statistical models. *IEEE Trans. Biomed. Eng.*, **55**(6):1754-1763. [doi:10.1109/TBME.2008.919128]

Zhao, Q.B., Zhang, L.Q., Cichocki, A., 2009. EEG-based asynchronous BCI control of a car in 3D virtual reality environments. *Chin. Sci. Bull.*, **54**(1):78-87. [doi:10.1007/s11434-008-0547-3]