Exploring Large Virtual Environments by Thoughts using a Brain-Computer Interface based on Motor Imagery and High-Level Commands

Fabien LOTTE[1,2,3,*], Aurélien VAN LANGHENHOVE[1,2,3],

Fabrice LAMARCHE[1,4], Thomas ERNEST[1,2,3], Yann RENARD[1,2],

Bruno ARNALDI[1,3], Anatole LÉCUYER[1,2,*]

[1]IRISA Rennes, France ; [2]INRIA Rennes, France

[3]INSA de Rennes, France ; [4]Université de Rennes 1, France

[*]Corresponding authors:

fabien.lotte@gmail.com, anatole.lecuyer@irisa.fr

Abstract

Brain-computer interfaces (BCI) are interaction devices which enable users to send commands to a computer by using brain activity only. In this paper, we propose a new interaction technique to enable users to perform complex interaction tasks and to navigate within large virtual environments (VE) by using only a BCI based on imagined movements (motor imagery). This technique enables the user to send high-level mental commands, leaving the application in charge of most of the complex and tedious details of the interaction task. More precisely, it is based on points of interest and enables subjects to send only a few commands to the application in order to navigate from one point of interest to the other. Interestingly enough, the points of interest for a given VE can be generated automatically thanks to the processing of this VE geometry. As the navigation between two points of interest is also automatic, the proposed technique can be used to navigate efficiently by thoughts within any VE. The input of this interaction technique is a newly-designed self-paced BCI which enables the user to send 3 different commands based on motor imagery. This BCI is based on a fuzzy inference system with reject options. In order to evaluate the efficiency of the proposed interaction technique, we compared it with the state-of-the-art method during a task of virtual museum exploration. The state-of-the-art method uses low-level commands, which means that each mental state of the user is associated to a simple command such as turning left or moving forwards in the VE. In

contrast, our method based on high-level commands enables the user to simply select its destination, leaving the application performing the necessary movements to reach this destination. Our results showed that with our interaction technique, users can navigate within a virtual museum almost twice as fast as with low-level commands, and with nearly twice less commands, meaning with less stress and more comfort for the user. This suggests that our technique enables to use efficiently the limited capacity of current motor imagery-based BCI in order to perform complex interaction tasks in VE, opening the way to promising new applications.

1 Introduction

BCI are communication devices which enable users to send commands to a computer application by using only their brain activity, this brain activity being measured and processed by the system (Wolpaw, Birbaumer, McFarland, Pfurtscheller, & Vaughan, 2002). Current BCI systems mostly rely on electroencephalography (EEG) for the measurement of this brain activity (Wolpaw et al., 2002). While BCI were initially designed as assistive devices for disabled people, they recently emerged as being also promising interaction devices for healthy users (Allison, Graimann, & Gräser, 2007). In this context, BCI have been notably used to interact successfully with virtual reality (VR) applications and video-games (Lécuyer et al., 2008). However, current techniques used for interacting with VE by thoughts remain relatively basic

and do not compensate for the small number of commands provided by the BCI. For instance, navigation techniques proposed so far are mostly based on low-level commands, with generally a direct association between mental states and turning left or right, or moving forwards in the VE.

We believe that higher-level commands should be more exploited in order to provide a more flexible, convenient and efficient system to interact with VE and video games. In such a system, most of the complex tasks would be carried out by the system whereas the user would only have to provide a few high-level commands to accomplish the desired tasks. Such a principle is currently being applied to brain-computer interactions with robots (Millán, Renkens, Mouriño, & Gerstner, 2004; Iturrate, Antelis, Minguez, & Kübler, 2009). Here, we rely on the fact that VE offer much more possibilities than the "real-life" conditions. For instance, it is very easy to add artificial stimuli or information in a VE and it is also much easier to extract information automatically from this VE, e.g., its topology, than from a real world environment. Therefore, we believe that a BCI-based interaction technique in VR should take advantage of these additional virtual possibilities.

In this paper, we present a novel interaction technique for exploring large VE by using a BCI based on motor imagery (i.e., imagined movements (Pfurtscheller & Neuper, 2001)), thanks to the use of high-level commands. Our technique is based on an automatic processing pipeline which can be used to interact by thoughts with any VE. The input of this interaction technique is a newly designed self-paced BCI based on 3 motor imagery classes. We

illustrate our interaction technique through the exploration of a virtual museum by thoughts, and compare it with the current technique used to explore VE with motor imagery-based BCI. Our results suggest that the use of high-level commands enables users to navigate the VE faster than with the current technique, based on low-level commands.

This paper is organized as follows: Section 2 proposes a state-of-the art of the existing interaction techniques used to explore VE by thoughts while Section 3 presents the interaction technique we proposed. Then, Section 4 presents the newly designed BCI system used as the interaction device. Section 5 describes the evaluation achieved and the obtained results. Finally, Sections 6 and 7 discuss the results and conclude, respectively.

2   Related work

Due to the appealing potential of BCI as a new input device, researchers have proposed an increasing number of 3D interaction techniques with VE based on BCI, with a special emphasis on navigation techniques (Lécuyer et al., 2008). In the work presented in this paper we mainly focus on navigation in VE by using a BCI. However, it should be mentioned that interesting works have been achieved on selection and manipulation of virtual objects with a BCI (Lalor et al., 2005; Bayliss, 2003).

In order to navigate in a VE by thoughts, most current works rely on brain signals generated by Motor Imagery (MI), i.e., brain signals generated

when the user imagines movements of his own limbs (e.g., hand or foot). Indeed, such mental tasks trigger variations of EEG signal power, in specific frequency bands, known as the μ (8-13Hz) and β (13-30 Hz) rhythms, over the motor cortex area of the brain (Pfurtscheller & Neuper, 2001). These variations are well studied and described in the literature, which makes them relatively easy to identify within a BCI. Moreover, MI does not require the use of external stimulus and as such enables the user to generate commands in a completely spontaneous way. When using a BCI based on MI, the detection of a given imagined movement is associated to a given control command.

BCI have been used to rotate the VE camera left or right by using imagination of left or right hand movement respectively (Friedman et al., 2007) or by using specific brain responses to 2 different visual stimuli (Touyama, 2008). It has also been used to walk and stop along a virtual street by using foot and right hand motor imagery (Leeb et al., 2006), to steer a virtual car left or right (Ron-Angevin & Diaz-Estrella, 2009) or to navigate in a maze (Ron-Angevin, Diaz-Estrella, & Velasco-Alvarez, 2009), still using MI. Leeb et al. have also proposed an interaction technique that enables users to explore a virtual apartment by thoughts (Leeb, Lee, et al., 2007). In this application, the user had to select at each junction the corridor (among two) he wanted to walk next, by using two MI tasks. Once the corridor selected, the user was moved automatically along predefined paths towards the next junction. However, it should be noted that in this application, only two commands were provided to the user and that the trajectories from one

6

junction to the other were manually defined. Moreover, if the user wanted to go to a distant junction, he could not select directly this location, but had to go through all the intermediate junctions. Recently, Guger et al. have also proposed an application to explore a virtual home using a P300-based BCI (Guger, Holzner, Grönegrass, Edlinger, & Slater, 2009). The P300 is a neurophysiological brain signal, known as an event related potential, that is triggered by a relevant and rare stimulus (Wolpaw et al., 2002). In this work, in order to navigate the VE, multiple visual stimuli were displayed to the user, each stimulus corresponding to a particular location in the virtual home. The user should focus on the stimulus corresponding to the place he wanted to go to, in order to actually move there. However, this system was not general-purpose in the sense that the stimuli used and the possible navigation locations were manually defined. As such, this system could not be used to automatically navigate a different VE. Finally, this system, together with the other interaction techniques described so far, are based on a synchronous BCI. These BCI can be used only during specific time periods, imposed by the system. Thus, with such a BCI, the user cannot send commands any time, which is not really convenient nor natural.

Very recently, some groups have started to investigate self-paced BCI, i.e., BCI that can be used to send commands at any time, at will (Leeb, Settgast, Fellner, & Pfurtscheller, 2007; Leeb, Friedman, et al., 2007; Mason, Kronegg, Huggins, Fatourechi, & Schloegl, 2006). Self-paced BCI can also recognize the so-called "non-control" mental state, i.e., they can recognize when the

7

user does not want to send a mental command (Mason et al., 2006). Using such self-paced BCI, researchers have enabled users to freely move forwards in a virtual street (Leeb, Friedman, et al., 2007) or to explore a virtual library by walking along predefined paths (Leeb, Settgast, et al., 2007). A self-paced BCI has also been used in an entertaining VR application in which the user could lift a virtual object up by using foot MI (Lotte, Renard, & Lécuyer, 2008).

Most interaction techniques described so far provided only a small number of commands to the user: generally two commands for the synchronous BCI mentioned above and a single command for the self-paced ones. Indeed, when using more mental states, i.e., more commands, the performance of the BCI, in terms of rate of correct mental state recognition, drops rapidly, hence generally compromising the reliability of the system (Wolpaw et al., 2002; Obermaier, Neuper, Guger, & Pfurtscheller, 2000). The most advanced technique for navigating a VE with a motor-imagery based BCI is currently the work of Scherer et al. In this work, the user could freely navigate in a virtual garden by using a self-paced BCI which provided 3 different commands (Scherer et al., 2008). In this application, users could turn left, turn right, or move forward in the VE by performing left hand, right hand or foot MI respectively. As an evaluation of this system, 3 subjects had to collect 3 coins in the VE, within a given time window. The results showed that 2 subjects out of 3 successfully completed the task (Scherer et al., 2008).

As a summary, even though the existing VR applications based on BCI are

8

already quite impressive, there is still room for improvement. In particular, most of the commands provided are low level commands, e.g., turn left or right. This leaves the user in charge of all the details of the navigation task, which can be tiring and uncomfortable. Moreover, with low level commands, each mental state used for control is associated to a fixed low-level command, hence restricting the number of interaction tasks offered to the user. The next section describes a new interaction technique based on high level commands, in order to overcome these current limitations. This technique uses as input a new self-paced motor imagery-based BCI providing the user with 3 different commands. The interaction technique is described here within the context of exploration of a virtual museum by thoughts.

3 A novel interaction technique to explore large VE based on a BCI using high-level commands

The targeted application for our interaction technique is an application in which a user could visit a large VE such as a virtual museum by using thoughts only. This application should enable the user to navigate in the virtual museum and to look at the artworks displayed. The targeted input device is a self-paced BCI which can provide its user with 3 commands, respectively associated to left hand, right hand and foot MI (see Section 4).

In order to provide the user with a flexible interface and several possible choices of tasks though using only 3 mental commands, we propose a new interaction technique which relies on a binary tree approach. This technique is described in the following.

3.1 Using binary decision trees for selecting interaction tasks

With our interaction technique, the tasks available to the user at a given instant are organized according to a binary tree structure. This means that the possible tasks are recursively divided into two subsets of tasks and are placed at the nodes and leaves of a binary tree. In order to select a specific task, the user should first select one of the two subsets of tasks displayed, by using either left hand MI (to select the first subset) or right hand MI (to select the second subset). This choice done, the selected subset of tasks is again divided into two subsets and displayed to the user who should take another choice. This means that the current node of the binary tree has been changed to the root node of the left or right subtree. This process is repeated until the selected subset contains only a single task (i.e., until a leaf of the binary tree is reached), which task is then carried out by the system. Contrary to existing BCI-based interaction techniques in VR, this principle enables the user to perform different kinds of interaction tasks, e.g., switching from a virtual object selection task to a navigation task. This is currently

10

impossible with other existing techniques for MI-based BCI.

In order to cope with human or BCI mistakes, BCI mistakes being much more common due to the modest performances of current mental state classifiers (Wolpaw et al., 2002), we also provide the user with an "undo" command. At any time, the user can perform foot MI in order to cancel the last choice he took and go back in the binary tree.

Based on this task selection principle, two navigation modes are provided to the user: the free navigation mode and the assisted navigation mode. The user can select one mode or the other by using left or right MI at the top of the binary tree (see Figure 1). When the user leaves a given mode by using the undo option (foot MI), the other mode is automatically selected in order to save time. Figure 1 displays the structure of this binary tree. As illustrated by this figure, it should be noted that this binary tree is composed of two parts: (1) a fixed part, used to switch between the two modes and to manage the free navigation mode, and (2) a dynamic part, constructed according to the user's viewpoint in the VE, and corresponding to the assisted navigation mode. The principle of this mode and the construction of the corresponding dynamic part of the tree are described in the next section.
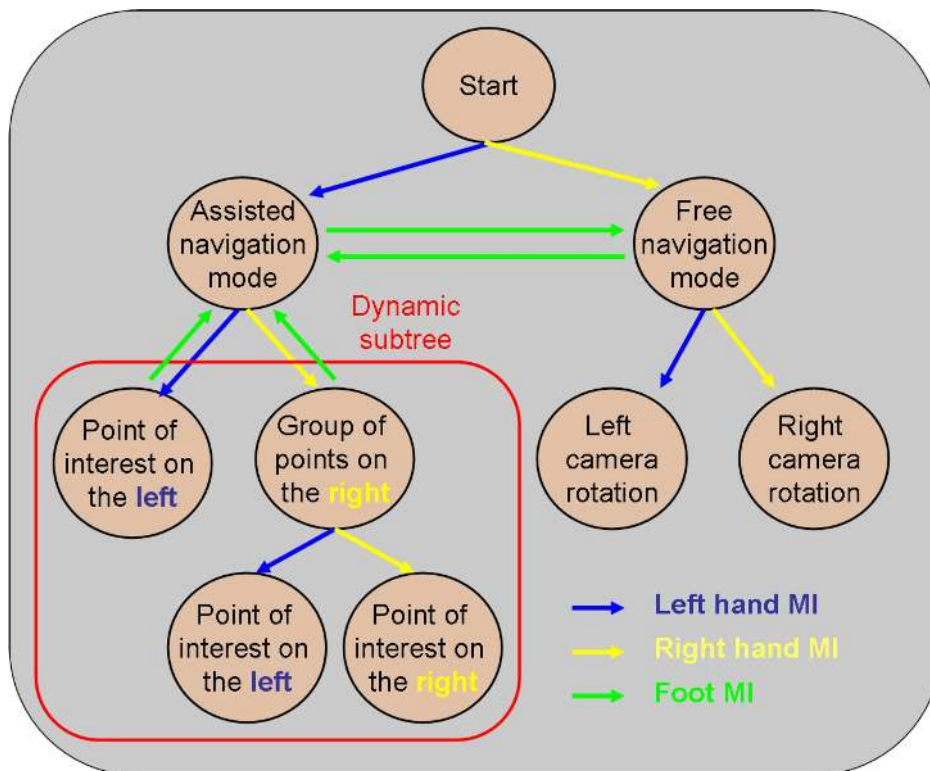
Figure 1: *Architecture of the binary tree used for task selection. It should be noted that the architecture of the subtree surrounded by a red line is dynamic. Indeed, the number of nodes in this subtree and their arrangement change automatically with the user's view point (see Section 3.2 for details).*

3.2 Assisted navigation mode: use of high-level commands

In the assisted navigation mode, the user can select points of interest by using the binary tree selection mechanism presented before. The points of interest that the user can select at a given instant depend on his position and field of view, as the user can only select visible points of interest. The

12

points of interest located the farthest on the left of the user's field of view are placed on the left subtree of the dynamic part of the binary tree (see Figure 1) whereas the points located the farthest on the right are placed on the right subtree. This process is recursively repeated until all the points of interest are arranged in the binary selection tree. As such, the user can naturally use left hand MI to select the set of points on the left, and right hand MI to select the set of points on the right. Naturally, this dynamic subtree is constructed every time the user moves, as this changes his field of view. Thus, it should be stressed that the size of the binary tree does not depend directly on the size of the VE but only on the number of points of interest that are currently visible by the user. This means that, with appropriately defined points of interest, the binary tree will remain relatively small, even for very large VE.

For the museum application, these points of interest could be either artworks or navigation points. Artworks represent any object exposed in the museum that is worthy of interest, such as a painting, a picture or a statue. Different kinds of interaction tasks could be proposed to the user according to the kind of artwork. For instance, concerning a painting, the user could need to focus and zoom on some parts of the painting in a 2D manner. On the other hand, concerning a statue, the user could need to turn around the statue to observe it from various points of view. While artworks can already be selected in our current system, it should be mentioned that artwork manipulation tasks are not implemented yet. However, the binary tree could be easily extended to handle them in the near future.

13

Navigation points are points that the user can navigate to. More precisely, in order to walk in the VE, the user just needs to select the next navigation point from the available navigation points, using the binary tree selection technique. The application automatically drives the walk from the current point to the selected one. This relieves the user from all the cumbersome details of the trajectory to go from one point to the other. During this assisted walk, the user can perform foot MI at any time (i.e., undo) in order to stop the walking where he is. This could be useful if the user changes his mind or if, during the walk, the user spots an interesting artwork, not visible previously. Interestingly enough, these navigation points can be generated automatically by our approach, by extracting topographical information from the geometry of the 3D model (see Section 3.5).

3.3 Free navigation mode: turning the view point left or right

This navigation mode is a simple mode which provides low-level commands to the user. It enables the user to rotate the camera towards the left or towards the right by using left or right hand MI respectively. This mode enables the user to look around its current position, in order to localize his next destination or to look for a given artwork in the case of the virtual museum application.

3.4 Graphical representation and visual feedback

Providing relevant feedback to any VR user is essential. This is particularly important for BCI-based applications as the user can only rely on the feedback to know whether the mental task was correctly detected by the BCI (Wolpaw et al., 2002). Moreover, results of a previous study have suggested that, for self-paced BCI-based VR applications, providing a continuous and informative feedback at any time may reduce the user's frustation and improve his learning (Lotte, Renard, & Lécuyer, 2008). Consequently, we pay attention to provide such a feedback to the users.

In our application, various colored icons are displayed dynamically on the screen (see Figure 2). Among these icons, 3 are continuously displayed: these icons provide feedback on the mental states identified by the system. These 3 icons represent a left hand (blue), a right hand (yellow) and feet (green) and are naturally associated to left hand, right hand and foot MI, respectively. When the VR application receives a given mental state from the BCI classifier, the size of the corresponding icon increases. As long as the same mental state is being received, the size of the icon keeps increasing until the number of consecutive states required is reached. Indeed, to make the control of the application more robust, we require that the same mental state is received several times in a row in order to execute the corresponding command. In other words, we used a dwell time. Dynamically changing the icon size depending on the BCI output enables us to provide feedback to the user even when the non-control state (any mental state except the targeted

15

MI states) is finally detected, as recommended by a previous study (Lotte, Renard, & Lécuyer, 2008). When the icon reaches its maximum size, the corresponding command is executed. This command is also represented on screen under the form of an icon placed next to the corresponding mental state icon, and displayed with the same color. This aims at informing the user of what will happen if he performs a given mental command. As the available commands depend on the mode used, the icons representing the commands are also dynamically changed.

Figure 2: *Graphical representation of the BCI-based virtual museum application and its associated interaction technique.*

The visible points of interest are displayed in the 3D scene using colored

16

pins (see Figure 2). When using the assisted navigation mode, the user can select these points to go automatically from one point to another. In this mode, the (set of) point(s) that can be selected using left hand MI are displayed in blue, i.e. with the same color as the left hand icon, whereas the point or the set of points that can be selected using right hand MI are displayed in yellow, i.e., with the same color as the right hand icon. The other points, which cannot be selected anymore, are displayed in red and black. Figure 3 displays an example of selection of navigation points, on which we can see these colored pins. When selecting these points of interest, no command icon is displayed on screen, as the points of interest are colored according to the mental command needed to select them.

3.5   Implementation

We have designed and implemented our application and interaction technique in such a way that it can be used with virtually any 3D model. In this section, we first present the algorithm used to generate the navigation points and to compute automatically trajectories between these points, i.e., to perform path planning. Then we present the general software architecture of the whole application.

3.5.1   Generation of navigation points and path planning

In order to generate navigation points and to automatically compute trajectories between these points, we used the TopoPlan (Topological Planner)
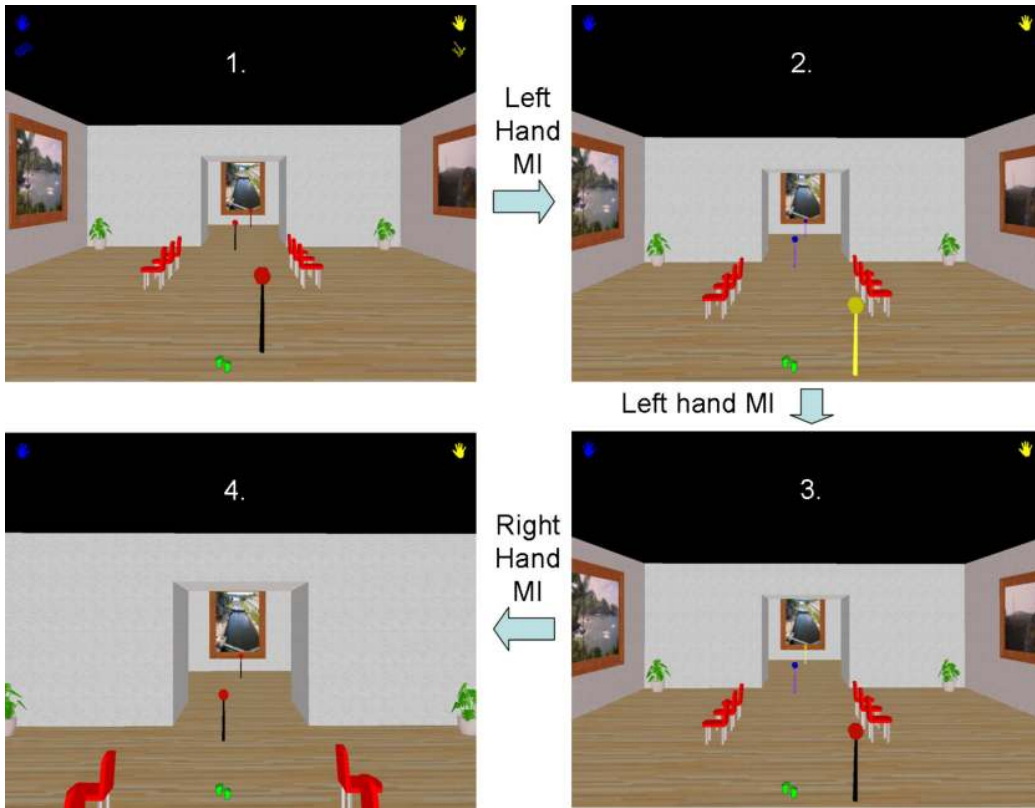
Figure 3: *Assisted navigation in the virtual museum. Picture 1: the user selects the assisted navigation mode by using left hand MI from the starting node of the binary tree. Picture 2: the user selects the set of two navigation points located on the left by using again left hand MI. Picture 3: from these two points, the user finally selects the one on the right (at the back) by using right hand MI. Picture 4: the application automatically drives the walk to the selected point.*

algorithm proposed by Lamarche (Lamarche, 2009). Using only the 3D geometry of a VE, Topoplan can extract a 3D topology of this environment. This topology defines the VE areas that can be navigated, i.e navigation points, and their accessibility. As such, Topoplan can be used to define a roadmap (a graph of navigation points) that enables any virtual entity to

go from one point of the environment to the other without encountering any obstacle. In other words, Topoplan can be used to perform automatic path planning (Lamarche, 2009).

Concerning the assisted navigation mode of our application, the challenge was to find a small number of relevant navigation points that enable the user to navigate the VE by selecting these points. To this end, we propose to filter the initial roadmap of navigation points. To do so, all the points from the roadmap are first given a mark. This mark corresponds to the distance between the point and the nearest obstacle. In other words, the highest a point mark, the more this point maximizes the coverage of the environment. The different points are then analysed by decreasing order of their associated mark. Let us define $S(p)$, a function which associates to each point p its successors in the graph representing the roadmap. A point $s \in S(p)$ is removed from the roadmap if and only if for all points $x \in S(s)$, the path (x, p) can be navigated. If a point s is removed, the paths (x, p) such that $p \in S(s)$ are added to the roadmap. This process is repeated until the algorithm converges. This algorithm enables us to filter the roadmap by keeping the points which maximize the visibility on the 3D scene in priority. As such, it can greatly simplify the roadmap while ensuring that there is at least another visible point which can be selected to navigate the VE. The navigation points we used within our interaction technique correspond to the points that have been selected after this filtering.

However, even if these points appeared as functional, some of them did

19

not have an optimal position, e.g., some of these were located in the corners of the rooms. Thus, after this automatic generation of points, we can still perform a manual optimization of their positions and eventually remove/add some points. However, it is worth noting that Topoplan can still generate automatic trajectories from point to point, should the points be generated automatically or by hand. As a result, any 3D model can be processed and used as input by our application

3.5.2 General software architecture

In order to use our BCI-based interaction technique, some offline operations are required beforehand. The main requirement is to have a 3D model of the VE. For our evaluations, we used the Google SketchUp modeling tool (Google, 2009) in order to create the 3D model of a virtual museum. In addition, an XML file, which contains the pathname of the artwork 3D models (if any) and the positions and orientations of these models in the museum, should be written. From the museum 3D model and the XML file, our application uses Topoplan to automatically generate the topology, the roadmap and the points of interest of the virtual museum. Optionally, these points of interest can be manually optimized. The BCI was implemented using the OpenViBE platform (INRIA, 2009). Within OpenViBE, a subject-specific BCI was calibrated for each user by using training EEG data from this user.

The main component of the software architecture is called the interaction engine. This software module processes the commands received from the BCI

in order to perform the corresponding interaction tasks. The interaction engine uses the Ogre 3D engine (Ogre3D, 2009) to render the VE and to display the visual feedback. Finally, the interaction engine uses Topoplan in order to perform the automatic navigation between two points of interest. The general software architecture is summarized in Figure 4.
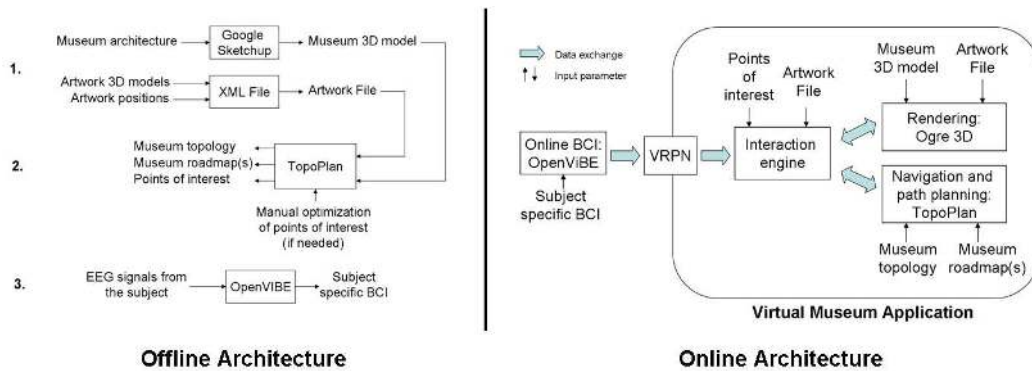

Figure 4: *General software architecture. Left: offline processing. Right: online processing.*

4   The BCI system

As mentioned earlier, our interaction technique is based on a self-paced BCI which can provide its user with 3 mental commands. As such, this BCI is a 4-state self-paced BCI. It can indeed recognize the 3 mental states associated to each command - the Intentional Control (IC) states - plus the Non-Control (NC) state, i.e., any mental state that does not correspond to a control command. The following sections describe the preprocessing, feature extraction and classification methods we used to design this self-paced BCI.

21

4.1 Preprocessing

In order to record the MI brain signals, we used 13 EEG electrodes, located over the motor cortex areas. These electrodes were FC3, FCz, FC4, C5, C3, C1, Cz, C2, C4, C6, CP3, CPz and CP4 according to the international 10-10 system (American Electroencephalographic Society, 1991) (see Figure 5). In order to preprocess the signals recorded by these electrodes, we first band-pass filtered the raw EEG signals in the 4-45 Hz frequency band as this frequency band is known to contain most of the neurophysiological signals generated by MI (Pfurtscheller & Neuper, 2001). Moreover, performing such a filtering can reduce the influence of various undesired effects such as slow variations of the EEG signal (which can be due, for instance, to electrode polarization) or power-line interference. To achieve this filtering, we used a Butterworth filter of order 4. In order to enhance the brain signals of interest, we also used a Surface Laplacian (SL) spatial filter (McFarland, McCane, David, & Wolpaw, 1997) over C3, C4 and Cz, leading to three Laplacian channels C3', C4' and Cz'. Features were extracted from these three new channels.

4.2 Feature extraction

For feature extraction we used logarithmic Band Power (BP) features as in (Pfurtscheller & Neuper, 2001). Extracting a band power feature consists in filtering the EEG signal of a given electrode in a given frequency band, squar-
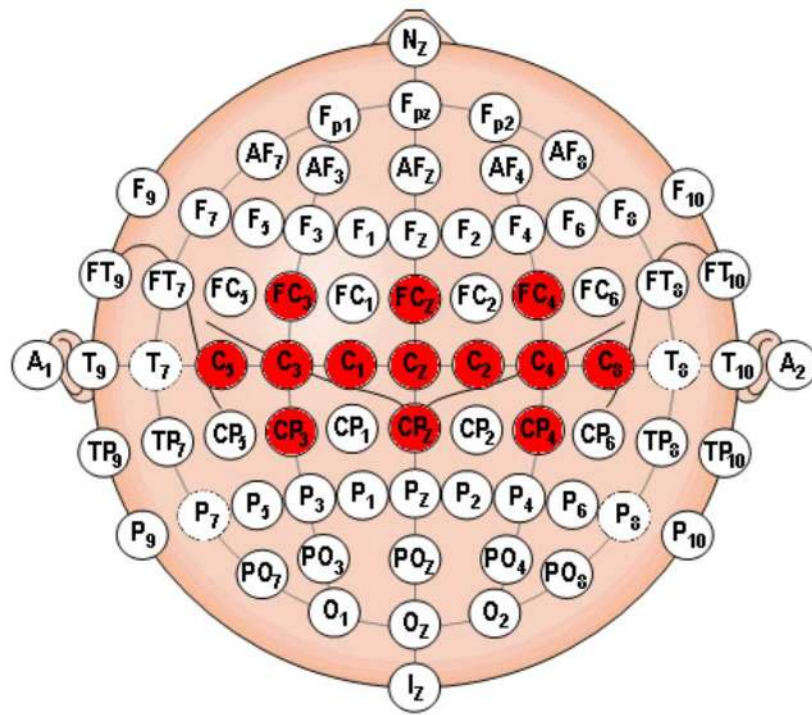
22

Figure 5: *Placement of electrodes in the 10-10 international system. The 13 electrodes we used in our BCI are displayed in red.*

ing the results and averaging it over a given time window, and finally taking its logarithm. Such a feature extraction is a popularly employed method which has been proved efficient for motor imagery-based BCI (Pfurtscheller & Neuper, 2001; Zhong, Lotte, Girolami, & Lécuyer, 2008). To obtain a more efficient BCI, we extracted several band power features in different frequency bands for the different Laplacian channels and selected the most relevant ones using the Sequential Forward Floating Search (SFFS) feature selection algorithm (Pudil, Ferri, & Kittler, 1994). This algorithm is indeed one of the most popular and efficient feature selection techniques (Jain & Zongker,

23

1997). More precisely, we investigated BP features extracted in 2 Hz wide frequency bands between 4 and 34 Hz, with a 1 Hz step, and selected the 12 most efficient features using the SFFS algorithm. We indeed observed that using more than 12 features did not increase significantly the performance whereas it increased the computational burden. As our BCI is self-paced, it requires continuous classification of EEG signals. Consequently, we extracted a BP feature vector 16 times per second, over the last 1 s time window (i.e., using a sliding window scheme).

4.3 Classification

For classification we used a Fuzzy Inference System (FIS) as described in (Lotte, Lécuyer, Lamarche, & Arnaldi, 2007). Such a classifier can learn fuzzy "if-then" rules from data, and use them to classify unseen data. This classifier has been reported to have many advantages for BCI, among which its efficiency for self-paced BCI design (Lotte, Mouchère, & Lécuyer, 2008). In order to design a self-paced BCI we relied on a pattern rejection approach. More precisely, we used the reject class technique, as this seemed to be the most efficient method according to results of a previous study (Lotte, Mouchère, & Lécuyer, 2008). With this technique, our FIS classifier will have to deal with 4 classes: one for each mental state used for control (Intentional Control states), and one class for all other mental states (Non Control state).

In order to make our self-paced BCI more robust, we also used a dwell time and a refractory period (Townsend, Graimann, & Pfurtscheller, 2004).

24

When using a dwell time, a given control command is generated only if the classification identifies the same class N_D times in a row. Similarly, when using a refractory period, the N_R classifications which immediately follow the identification of an IC state will be forced to the NC state. These two techniques lead to less False Positives (FP), i.e., to less identifications of an IC state instead of an NC state. In our system we used $N_D = N_R = 7$. These values were defined experimentally according to users' preferences.

This new BCI design enables the user to employ 3 different commands in a self-paced manner. So far, very few groups have proposed a self-paced BCI with that many commands, exceptions being (Scherer et al., 2008) and (Bashashati, Ward, & Birch, 2007). Moreover, according to previous studies (Lotte, Mouchère, & Lécuyer, 2008), this new design is expected to have better performances than these existing designs.

5   Evaluation

The aim of the evaluation was to identify whether the interaction technique we proposed was usable and efficient. Additionally, it was interesting to assess the performance of the BCI and the users' experience. To do so, we studied the performances of 3 participants who used our interaction technique to navigate from room to room in a virtual museum. As a comparison, participants also had to perform the same task by using the state-of-the-art

25

method, i.e., turning left, turning right or moving forwards by using left hand, right hand and foot MI, respectively, as in (Scherer et al., 2008). This aimed at comparing our high-level approach to a low-level one. Naturally, in order to perform a fair comparison, we used the same BCI design (the BCI presented in Section 4) for both interaction techniques. This BCI design is different from the one used by Scherer et al. (Scherer et al., 2008), as we used a different classification algorithm, different electrodes, different signal processing techniques, etc. However, according to previous studies (Lotte, Mouchère, & Lécuyer, 2008), the algorithms we used should give better recognition performances than the BCI design used originally by Scherer et al. Moreover, in this paper we mainly aimed at comparing the interaction techniques, but not the BCI designs. It should also be mentioned that we did not use a refractory period with the low-level interaction technique. Indeed, this enabled the user to maintain the MI task in order to maintain the movement in the museum and as such to move continuously rather than by periods. When using the low-level technique, the user moved by about 25 cm for each classifier output labelled "foot MI" (forward movement), knowing that the classifier provides 16 outputs per second. In other words, with the low level approach, the user could move at up to 4 m per second (16×0.25). When using the high-level technique, participants moved at the speed of about 2.5 m per second during the automatic navigation phases. Thus, it should be mentioned that the movement speed was set slightly lower with the high-level technique. Indeed, fast passive movements may be visually

26

uncomfortable for the user.

As a measure of performance, we evaluated the time needed by each participant to perform different navigation tasks. As this evaluation was dedicated to navigation tasks, artworks were not considered during this experiment, which means users could not interact with them. However, artworks were still displayed in order to provide an engaging VE. The next sections describe the virtual museum used, the population and apparatus, the task participants had to do, the experimental procedure and finally the results obtained.

5.1   Virtual museum

For this evaluation, we used a fictional virtual museum. This museum was composed of 8 different rooms each one containing either several pictures or a statue (see Figure 6).

5.2   Population and apparatus

For the experiment, EEG signals were recorded using a Nexus 32b EEG machine from the Mind Media company, at a sampling frequency of 512 Hz. Three participants took part in this experiment (S1,S2 and S3, all males, mean age: 25.7±2.11), two of which had a previous MI-based BCI experience (S1 and S2). The experimental setup is displayed on Figure 7.
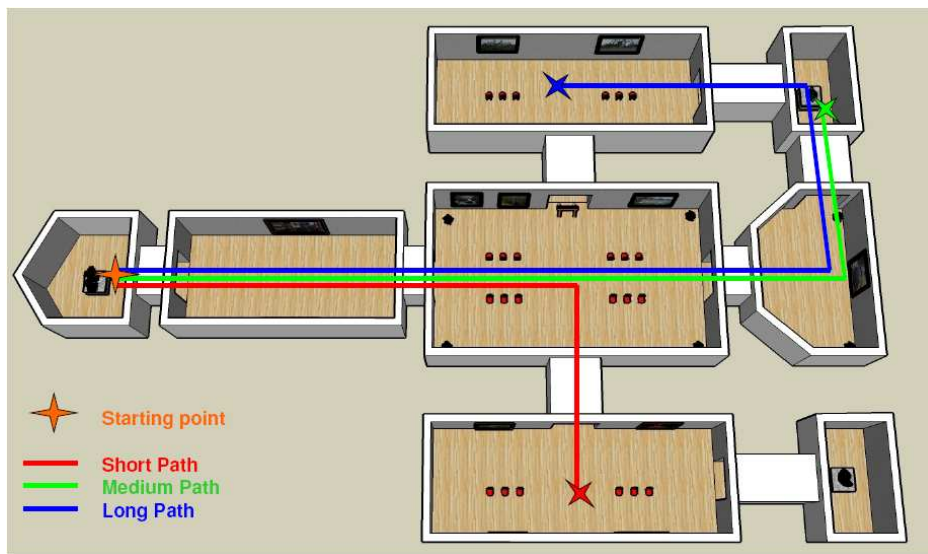
Figure 6: *3D model of the virtual museum used in our experiment on which some paths followed by the participants (short, medium, long) are visible.*

5.3   Navigation tasks

For this evaluation, participants had to navigate from one room to another as fast as they could. There were three possible distances to cover: short, medium and long paths. These tasks consisted in passing through 3, 4 or 5 rooms respectively (see Figure 6 for examples).

5.4   Procedure

Before taking part in the virtual museum experiment, each participant performed three to five sessions following the training protocol of the 3-class Graz BCI (Pfurtscheller & Neuper, 2001), in order to record training data for selecting the features and training the FIS classifier. These sessions were
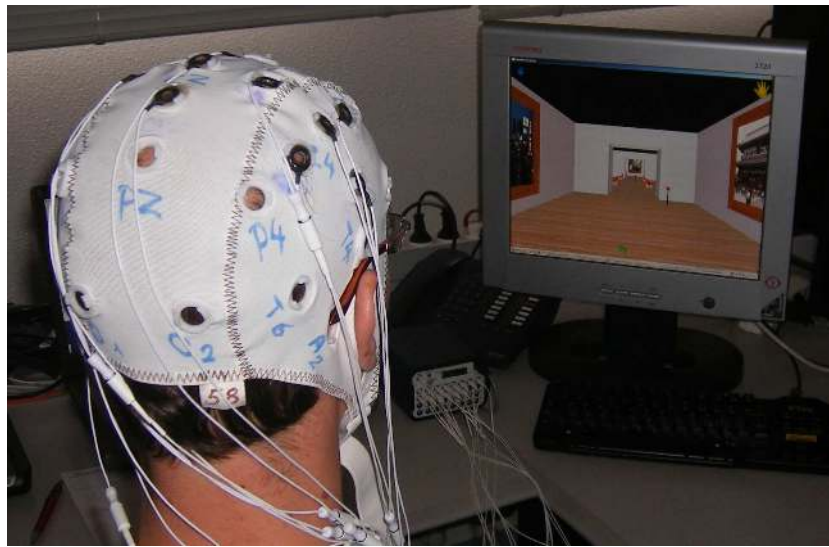
Figure 7: *Setup of the experiment.*

recorded on a different day than the virtual museum experiment. During a session, the participant had to perform 15 trials for each class (left hand, right hand and foot MI) i.e., 15 repetitions of each mental state for a duration of 4 seconds. As no training classifier was available for these first sessions, the participant was not provided with any feedback. The participant was also instructed not to perform any MI nor real movement during the inter trial periods as the data in these periods are used as examples of the NC state. Once these sessions were completed, the features were selected and the classifier was trained on these data.

Each participant had to perform each kind of navigation task twice for each of the two interaction techniques. The order of the tasks was arranged in blocks. Within a block, participants used a single interaction technique and

had to perform once each kind of navigation task (short, medium and long - the actual order being randomized within the block). The order of the blocks was also randomized. These blocks were achieved over different days as the tasks were too tiring for the participants to be carried out on a single day. Each day, the duration of the experiment was approximately 2 to 2.5 hours, including the time required for electrode montage. At the beginning of each day, the participant participated in three sessions of the training protocol of the 3-class Graz BCI (see above). During these sessions, the participant was provided with a feedback thanks to a classifier trained on data recorded during the previous day of experiment. It should be noted that the BCI used for these sessions was synchronous, which means that the NC state was not recognized by the BCI. However, participants were instructed not to perform any MI nor real movement during the inter trial periods as the data in these periods would be used as examples of the NC state for training the final self-paced BCI. Once the three sessions were completed, the classifier was re-trained on these new data, in order to adapt to the participant's current EEG signals and in order to design the self-paced BCI. Then the experiment with the virtual museum could begin. After the experiments were completed, subjects were asked to fill in a subjective questionnaire.

5.5 Results

As mentioned above, the main purpose of the evaluation was the efficiency of the interaction technique we proposed. In the mean time, we assessed

the performance of the BCI system we designed and recorded the users'
experience. The results of these different evaluations are reported below.

5.5.1 BCI performance

First, it should be mentioned that all participants managed to reach all the
rooms they were instructed regardless of the interaction technique used. This
suggests that participants could actually control the application and explore
the virtual museum by using the proposed BCI system. As the BCI is self-
paced and as participants send commands at their free will, it is not possible
to report on the online classification performances of the BCI. Indeed, we can-
not compare the mental state recognized by the BCI with the mental state
performed by the participants as we did not know what the participants in-
tented to do at a given instant. However, in order to evaluate whether our
BCI system works better than a randomly performing BCI (i.e., a BCI which
is unable to properly identify any mental state), we simulated a random BCI.
More precisely, we sent randomly selected mental states to the application
as if it was the mental state identified by the classifier. We performed these
simulations twice for each interaction technique, the "objective" of the ran-
dom classifier being to perform a short navigation task. For each task, even
after 30 minutes of simulation (i.e., 1800 seconds), the random BCI was not
able to reach the targeted room. By comparison, the average time required
to go from one room to another (situated at a distance of 3 rooms or more)
with our BCI, independently from the interaction technique or the navigation

task, was 331.5 seconds, i.e., approximatly 5.5 minutes (see tables below). This suggests that our BCI indeed provided control to the user.

5.5.2 Interaction technique efficiency

Table 1 displays the average time needed by each participant to accomplish the different navigation tasks (short, medium or long paths), according to the interaction technique.

Table 1: *Average time (in seconds) needed by each participant to accomplish the different navigation tasks using the two interaction techniques.*

interaction technique	navigation task	S1	S2	S3	mean	overall mean
High-level commands	long	158	381.9	157.6	232.5	232.01
	medium	204.5	640.6	146	330.37	
	short	135	151.4	113.1	133.17	
Low level commands	long	702	530.5	598.1	610.2	433.84
	medium	554.5	598.7	336.8	496.67	
	short	235	85.7	263.3	194.67	

These results show that, in this study, navigating from one room to another by using high-level commands is on average almost twice as fast as when using low-level commands. It should also be reminded that the movement speed was slightly lower with high-level commands than with low-level commands (see Section 5), which further emphasizes the difference in efficiency. This difference is logically smaller for short paths and larger for long paths. A paired t-test (Norman & Streiner, 2008) com-

paring the time needed to navigate using high-level commands and using low-level commands, over all blocks, navigation tasks, and participants (i.e., with $n = 3$ *participants* $\times$ 2 *blocks* $\times$ 3 *navigation tasks* $= 18$ measures for each type of interaction technique), revealed that high-level commands were significantly faster ($p < 0.001$). In order to also investigate the presence of interactions between the technique used and the distance covered, we also performed a two-way ANOVA (Norman & Streiner, 2008) with factors *technique* (levels: low-level commands and high-level commands) and *distance* (levels: short, medium and long). The results revelead a significant effect for both factors *technique* and *distance* ($p < 0.01$), but no significant interaction *technique* $\times$ *distance* ($p = 0.19$). However, this lack of significant interaction might be due to the small sample size.

The time needed to navigate was of course related to the number of commands that participants had to send. Table 2 displays the average number of commands needed by each participant to accomplish the different navigation tasks, according to the interaction technique.

This table shows that participants sent on average 67 commands to go from the starting room to the destination room using low level commands, and only 36 commands to do the same thing using high-level commands. Thus, these results suggested that by using high-level commands, users need to send to the application almost twice as few commands as with low-level commands, hence performing their navigation tasks faster. However, it should be mentioned that a paired t-test revealed no significant difference

Table 2: *Average number of commands needed by each participant to accomplish the different navigation tasks using the two interaction techniques.*

interaction technique	navigation task	S1	S2	S3	mean	overall mean
High-level commands	long	32.5	43	24	33.17	36.11
	medium	33	76.5	44	51.17	
	short	20	26.5	25.5	24	
Low-level commands	long	142	53	97.5	97.5	66.89
	medium	131	32.5	63	75.5	
	short	19	10.5	53.5	27.67	

($p = 0.1$) between the number of commands needed with each technique. Again, this lack of significance might be due to the small sample size.

5.5.3 Users' experience

The analysis of the questionnaires shown that all subjects found navigation with points of interest less tiring than low-level navigation, as they could relax and rest during the automatic navigation periods from one point of interest to the other. Interestingly enough, a user reported that, with low-level commands, he sometimes stopped sending mental commands to rest a bit, as continuously performing MI was too tiring for him. In contrast, with high-level commands, this user reported that he used the automatic navigation phase to rest. In the questionnaire, participants also reported that the advantage of using an assisted navigation was particularly striking when an obstacle (e.g., a statue) was present in the path. Indeed, with

34

the low-level technique, many commands were necessary to walk round the obstacle whereas when using high-level navigation in the same situation, the application avoided the obstacle on its own, without additional demand on the user.

Overall, the questionnaire revealed that participants have found navigation using points of interests more efficient and generally as much as or more comfortable than navigation based on low-level commands. On the other hand, questionnaires revealed that using points of interest was less simple and less intuitive than using low-level commands. In other words, it seems that the increase in efficiency and comfort obtained by using high-level commands was gain at the expense of a more complex and less natural navigation. However, these latter issues seemed to vanish after some pratice and training, as the participants finally got used to this high-level technique. Moreover, participants reported that, between the two techniques, they all prefered using the one based on high-level commands.

6   Discussion

As expected, navigating using points of interest enables participants to send only a few high-level commands while leaving most of the work to the application itself. It should be stressed that this application is independent of the BCI. High-level commands enable users to navigate VE more rapidly by thoughts, especially when the VE is large and the path is long. Indeed, it

takes approximatly the same time to select close and far points of interest, provided that they are both visible. Leaving the application in charge of performing the low-level parts of the interaction tasks also enables the user to perform only a limited number of mental tasks as well as to rest. This makes the whole technique more comfortable and less tiring for the user, as reported by the subjects during the experiments. Indeed, when using low-level techniques to navigate, the user generally has to send mental commands continuously, which can become rapidly exhausting.

Although only 3 participants participated in this experiment, we should stress that this is a typical number of participants for current BCI experiments in VR, as these experiments are tedious, long and demanding for the participants (Leeb et al., 2006; Scherer et al., 2008; Friedman et al., 2007).

A limitation of our approach is that it prevents users from going everywhere as they can only go from navigation point to navigation point. However, it is still possible to add more points of interest in order to increase the user's mobility. We could also grant the user with additional low level navigation techniques. Indeed, thanks to the binary tree selection principle, any number of new modes or tasks can be easily added to the interface. For instance, this may enable the use of two techniques successively: high-level commands to reach a far destination quickly and low-level commands for precise and final fine-grain positionning.

It is also worth noting that any other 3D scene could be used with our approach. Indeed, generating the navigation points and performing path

planning can be done completely automatically using only the geometry of the VE, thanks to Topoplan (Lamarche, 2009).

The proposed system is based on Motor Imagery (MI). Other brain signals could have been used to drive this application, such as the P300 (Wolpaw et al., 2002). Indeed, some recent works have reported applications using a P300-based BCI to navigate in a virtual home (Guger et al., 2009) or in real environments (for wheelchair control) (Rebsamen et al., 2007; Iturrate et al., 2009), by selecting a destination among several.

In our opinion, both P300 and MI have pros and cons. The P300 can be used without any (human) training and it can be used to select among many targets (typically more than 3) with high information transfer rates. On the other hand, the P300 relies on external stimuli, these stimuli being generally visual. As such, the resulting BCI is necessarily synchronous in the sense it is dependent on the stimuli presentation. Moreover, these visual stimuli can be annoying or uncomfortable for the user, as it usually consists of flashing stimuli. Finally, it requires the user's attention to be focused on these stimuli, which prevents him from drawing its attention on other parts of the real or virtual environment. For instance, in the works presented in (Guger et al., 2009; Rebsamen et al., 2007; Iturrate et al., 2009), the user had to look at a separate screen which was in charge of providing the visual stimulus, hence preventing him from observing simultaneously the environment. This can be unconvenient when the environment itself is visually interesting or informative, e.g., as in our application with a virtual museum and artworks

to observe.

On the contrary, MI-based BCI are not based on any stimulus. As such, they can be used to send command to the application at any time, when the user decides to. Moreover they leave the visual attention of the user available, hence allowing him to look at the environment he is exploring, possibly while sending mental commands at the same time. Finally, MI can be seen as a natural and intuitive mental task to use. However, when used to select among several targets, MI-based BCI are most probably slower than P300-based BCI. MI-based BCI may also require training for the user, although it has be shown that with appropriate signal processing and classification techniques this training can be reduced or suppressed (Blankertz, Dornhege, Krauledat, Curio, & Müller, 2007).

In this paper, we have explored the use of MI-based BCI for exploring VE. However, we believe that a promising extension of the current work could be to use a hybrid BCI based on both MI and the P300. For instance, the part of our interaction technique based on visual information, such as the selection of points of interest, could be achieved using the P300. In that case, the points of interest could be randomly flashing, and the user would select one by focusing its visual attention on it. Other interaction tasks, e.g., rotating the camera towards the left or right, cancelling the last task performed, or interacting with an artwork (which requires the user's attention to be available) could be achieved using MI. This will be explored in a future work.

7 Conclusion

In this paper, we have presented a new interaction technique to explore large Virtual Environments (VE) by using a Brain-Computer Interface (BCI) based on Motor Imagery (MI). This technique is based on points of interest and enables participants to send only a few high-level commands to the application in order to navigate from one point of interest to the other. The actual navigation between two points of interest is carried out by the application itself (which is independent of the BCI), leaving the user free to rest or to observe the VE. Interestingly enough, the computation of these points and the navigation between these points can be achieved completely automatically. In order to select these points of interest, the user can employ a selection mechanism which relies on a binary tree and depends on the user's viewpoint. This binary tree mechanism provides a unified framework for BCI-based interaction in VR. Indeed, it enables, for instance, to select navigation commands or object manipulation commands in the same way. In addition, we also proposed a new design of a self-paced BCI which can issue three different commands, these commands being associated to left hand, right hand and foot MI, respectively. This BCI design is based on a fuzzy inference system with a reject option.

An evaluation of our system has been conducted within the framework of a virtual museum exploration. Results suggest that our method was efficient as, using our technique based on high-level commands, a user could navi-

gate the museum almost twice as fast as when using low-level commands. Moreover, our technique appeared as less tiring and more comfortable than the low-level approach. This suggests that our whole system may be used to navigate and interact with virtual worlds more efficiently and with more degrees of freedom than current approaches based on motor imagery. Thus, the proposed approach might enable both healthy and disabled users to interact seamlessly with the same virtual world, thanks to a BCI.

Future work could deal with evaluating this application with disabled people, in order to assess to which extent the application can enable them to visit virtually various museums or even other buildings which they might not be able to see for real. It should also be interesting to further improve the self-paced BCI in order to enable the user to navigate more easily and faster. For instance, we could use more advanced feature extraction techniques such as Common Spatial Patterns (CSP) (Blankertz, Tomioka, Lemm, Kawanabe, & Müller, 2008) or inverse solutions based algorithms (Lotte, Lécuyer, & Arnaldi, 2009). Finally, we could explore and add various interaction commands to our binary tree mechanism. For instance, interaction techniques such as Navidget (Hachet, Décle, Knödel, & Guitton, 2008) could be adapted for BCI purposes in order to easily observe the museum virtual statues.

Acknowledgments

References

Allison, B., Graimann, B., & Gräser, A. (2007). Why use a BCI if you are healthy ? In *proceedings of brainplay 2007, playing with your brain.*

American Electroencephalographic Society. (1991). Guidelines for standard electrode position nomenclature. *J Clin Neurophysiol.*, *8*(2), 200-202.

Bashashati, A., Ward, R., & Birch, G. (2007). Towards development of a 3-state self-paced brain-computer interface. *Computational Intelligence and Neuroscience.*

Bayliss, J. D. (2003). The use of the P3 evoked potential component for control in a virtual apartment. *IEEE Transactions on Neural Systems and Rehabilitation Engineering, 11*(2), 113-116.

Blankertz, B., Dornhege, G., Krauledat, M., Curio, G., & Müller, K.-R. (2007). The non-invasive Berlin brain-computer interface: Fast acquisition of effective performance in untrained subjects. *NeuroImage, 37*(2), 539-550.

Blankertz, B., Tomioka, R., Lemm, S., Kawanabe, M., & Müller, K.-R.

(2008). Optimizing spatial filters for robust eeg single-trial analysis. *IEEE Signal Proc Magazine*, *25*(1), 41-56.

Friedman, D., Leeb, R., Guger, C., Steed, A., Pfurtscheller, G., & Slater, M. (2007). Navigating virtual reality by thought: What is it like? *Presence*, *16*(1), 100-110.

Google. (2009). *Google sketchup website.* (http://sketchup.google.com/)

Guger, C., Holzner, C., Grönegrass, C., Edlinger, G., & Slater, M. (2009). Brain-computer interface for virtual reality control. In *Proceedings of esann 2009* (p. 443-448).

Hachet, M., Décle, F., Knödel, S., & Guitton, P. (2008). Navidget for easy 3D camera positioning from 2D inputs. In *Proceedings of ieee 3dui - symposium on 3d user interfaces.*

INRIA. (2009). *Open-ViBE platform website.* (http://openvibe.inria.fr/)

Iturrate, I., Antelis, J., Minguez, J., & Kübler, A. (2009). Non-invasive brain-actuated wheelchair based on a P300 neurophysiological protocol and automated navigation. *IEEE Transactions on Robotics*, *25*(3), 614-627.

Jain, A., & Zongker, D. (1997). Feature selection: evaluation, application, and small sample performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*(2), 153-158.

Lalor, E., Kelly, S. P., Finucane, C., Burke, R., Smith, R., Reilly, R., et al. (2005). Steady-state VEP-based brain-computer interface control in an immersive 3-D gaming environment. *EURASIP journal on applied signal processing.*

Lamarche, F. (2009). Topoplan: a topological path planner for real time human navigation under floor and ceiling constraints. In *Eurographics'09* (p. 649-658).

Lécuyer, A., Lotte, F., Reilly, R., Leeb, R., Hirose, M., & Slater, M. (2008). Brain-computer interfaces, virtual reality and videogames. *IEEE Computer*, *41*(10), 66-72.

Leeb, R., Friedman, D., Müller-Putz, G. R., Scherer, R., Slater, M., & Pfurtscheller, G. (2007). Self-paced (asynchronous) BCI control of a wheelchair in virtual environments: A case study with a tetraplegic. *Computational Intelligence and Neuroscience, special issue: "Brain-Computer Interfaces: Towards Practical Implementations and Potential Applications", 2007.*

Leeb, R., Keinrath, C., Friedman, D., Guger, C., Scherer, R., Neuper, C., et al. (2006). Walking by thinking: The brainwaves are crucial, not the muscles! *Presence: Teleoperators and Virtual Environments*, *15*(5), 500-51.

Leeb, R., Lee, F., Keinrath, C., Scherer, R., Bischof, H., & Pfurtscheller, G. (2007). Brain-computer communication: Motivation, aim and impact of exploring a virtual apartment. *IEEE, Transactions on Neural Systems & Rehabilitation Engineering.*

Leeb, R., Settgast, V., Fellner, D., & Pfurtscheller, G. (2007). Self-paced exploration of the austrian national library through thought. *International Journal of Bioelectromagnetism*, *9*(4), 237-244.

43

Lotte, F., Lécuyer, A., & Arnaldi, B. (2009). FuRIA: An inverse solution based feature extraction algorithm using fuzzy set theory for brain-computer interfaces. *IEEE transactions on signal processing, 57*(8), 3253-3263.

Lotte, F., Lécuyer, A., Lamarche, F., & Arnaldi, B. (2007). Studying the use of fuzzy inference systems for motor imagery classification. *IEEE Transactions on Neural System and Rehabilitation Engineering, 15*(2), 322-324.

Lotte, F., Mouchère, H., & Lécuyer, A. (2008). Pattern rejection strategies for the design of self-paced eeg-based brain-computer interfaces. In *International conference on pattern recognition (icpr).*

Lotte, F., Renard, Y., & Lécuyer, A. (2008). Self-paced brain-computer interaction with virtual worlds: a qualitative and quantitative study 'out-of-the-lab'. In *4th international brain-computer interface workshop and training course* (p. 373-378).

Mason, S., Kronegg, J., Huggins, J., Fatourechi, M., & Schloegl, A. (2006). *Evaluating the performance of self-paced BCI technology* (Tech. Rep.). Neil Squire Society.

McFarland, D. J., McCane, L. M., David, S. V., & Wolpaw, J. R. (1997). Spatial filter selection for EEG-based communication. *Electroencephalographic Clinical Neurophysiology, 103*(3), 386-394.

Millán, J., Renkens, F., Mouriño, J., & Gerstner, W. (2004). Noninvasive brain-actuated control of a mobile robot by human EEG. *IEEE Trans-*

actions on *Biomedical Engineering*, *51*(6), 1026-1033.

Norman, G., & Streiner, D. (2008). *Biostatistics: The bare essentials (3rd edition)*. B.C. Decker.

Obermaier, B., Neuper, C., Guger, C., & Pfurtscheller, G. (2000). Information transfer rate in a five-classes brain-computer interface. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *9*(3), 283-288.

Ogre3D. (2009). *Ogre 3d website.* (http://www.ogre3d.org/)

Pfurtscheller, G., & Neuper, C. (2001). Motor imagery and direct brain-computer communication. *proceedings of the IEEE*, *89*(7), 1123-1134.

Pudil, P., Ferri, F. J., & Kittler, J. (1994). Floating search methods for feature selection with nonmonotonic criterion functions. In *Pattern recognition, proceedings of the 12th iapr international conference on computer vision & image processing* (Vol. 2, p. 279-283).

Rebsamen, B., Burdet, E., Guan, C., Zhang, H., Teo, C. L., Zeng, Q., et al. (2007). Controlling a wheelchair indoors using thought. *IEEE Intelligent Systems*, *22*(2), 18-24.

Ron-Angevin, R., & Diaz-Estrella, A. (2009). Braincomputer interface: Changes in performance using virtual reality technique. *Neuroscience Letters*, *449*(2), 123-127.

Ron-Angevin, R., Diaz-Estrella, A., & Velasco-Alvarez, F. (2009). A two-class brain computer interface to freely navigate through virtual worlds. *Biomedizinische Technik/Biomedical Engineering*, *54*(3), 126-133.

45

Scherer, R., Lee, F., Schlogl, A., Leeb, R., Bischof, H., & Pfurtscheller, G. (2008). Towards self-paced brain-computer communication: Navigation through virtual worlds. *IEEE, Transactions on Biomedical Engineering, 55*(2), 675-682.

Touyama, H. (2008). Advances in human computer interaction. In (Shane Pinder ed., chap. Brain-CAVE Interface Based on Steady-State Visual Evoked Potential). InTech Education and Publishing.

Townsend, G., Graimann, B., & Pfurtscheller, G. (2004). Continuous EEG classification during motor imagery-simulation of an asynchronous BCI. *IEEE Transactions on Neural Systems and Rehabilitation Engineering, 12*(2), 258-265.

Wolpaw, J., Birbaumer, N., McFarland, D., Pfurtscheller, G., & Vaughan, T. (2002). Brain-computer interfaces for communication and control. *Clinical Neurophysiology, 113*(6), 767-791.

Zhong, M., Lotte, F., Girolami, M., & Lécuyer, A. (2008). Classifying EEG for brain computer interfaces using gaussian processes. *Pattern Recognition Letters, 29*, 354-359.