Accurate Visual Stimulus Presentation Software for EEG Experiments

Malypoeur Plong[12], Kai Shen[1], Marijn Van Vliet[3], Arne Robben[3], Marc Van Hulle[3], Luc Geurts[1]
[1]e-Media Lab, GROUP T – Leuven Engineering College, Leuven, Belgium
[2]Computer Science, Royal University of Phnom Penh, Phnom Penh, Cambodia
[3]Computational Neuroscience Group, KU Leuven, Leuven, Belgium

In many experimental paradigms of electroencephalography (EEG), the timing accuracy in presenting experimental stimuli as visualized on a computer screen is of great importance. This paper describes a software package enabling students and researchers to easily set up suitable visual stimulation sequences for simple experiment. We describe the mechanisms employed by the software to achieve good timing accuracy on a common laptop, running Microsoft Windows. With a light sensor, the synchronization of the visual stimulation and the EEG hardware is evaluated. The results demonstrate a maximum variance of 2ms between the synchronization marker sent to the EEG hardware and the presentation of the stimulus on screen. We also confirm the software's ability to display a flickering stimulus, used in steady-state visual evoked potential (SSVEP) experiments, with a precise frequency.

Keywords: Accurate timing, Flickering stimulus, EEG experiments support

1. INTRODUCTION

We have developed a software package to support our students and researchers in the field of brain-computer interfacing (BCI).

A BCI is a communication system in which messages or commands that an individual sends to the external world do not pass through the brain's normal output pathways of peripheral nerves and muscles [1]. This technology aims to provide a mean for people with motor disabilities to express their wishes and be able to perform some actions by using their brain signal. Signals from the brain are detected and analyzed by computer algorithms. The analysis of these signals can reveal the intention of the user, enabling him or her to give commands to a computer and to control an output device.

A common way for BCI to measure brain activity is through encephalography (EEG), where electrodes are placed along the scalp. People, who are working with BCI, will most of the time have to perform some simple EEG experiments before implementing a full BCI. We, therefore, developed an easy-to-use program for designing these experiments, which specifically aim to easily present visual stimuli on an LCD screen with precise timing.

1.1 EEG-BASED BCI PARADIGMS

BCIs are divided into different groups based on the electrophysiological signal they use such as VEPs, SSVEPs, P300, Slow Cortical Potentials, and β and μ rhythms [1]. These paradigms are used for different purposes but the ones that receive the most attention in our lab are P300 evoked potentials and SSVEP.

P300 Evoked Potential: This potential is a component of an event related potential (ERP) that is the measured brain response after presenting stimulus. Stimulus can be auditory, visual or somatosensory. This particular wave has been used to detect the subject's choice between different stimuli [2][3]. The P300 is a component of the ERP that is usually elicited using an oddball experiment in which low-probability target items (stimuli desired by the user e.g. a letter, an icon) are mixed with high-probability non-target items (the stimuli not desired by the user). Focusing on the target items will elicit the P300 component [1].

Steady-State Visual Evoked Potentials (SSVEP): These signals are the responses to visual stimuli flickering at a rate higher than 6Hz [4]. The individual transient visual response overlap leads to a steady state signal that resonates at the same rate as the stimulus and its multipliers. It can be used to detect a user's choice when presenting him or her the stimuli with different flashing frequencies [5][6][7]. In a BCI system, several commands might be simultaneously presented on screen. Each command flashes at a fixed rate but different frequencies. Gazing at each of the targets will result in different SSVEP responses. Decoding these responses will then lead to control (selection) of the targets.

1.2 ACCURATE TIMING IN EEG EXPERIMENTS

In order to analyze the EEG signal, it must be synchronized with the stimulus presentation on the computer screen. In the case of the P300 paradigm, an ERP will be generated after every presence of the target stimulus. To detect the P300, multiple ERPs will be aligned and averaged. If the alignment is not precise, P300 peaks will be smeared across time. Thus, in order to have an accurate and correct detection of user's intent in the P300 paradigm, it is important to know the exact time instant when the stimulus appears on the screen.

In case of the SSVEP paradigm, the brain generates electrical activity at the same frequency as the chosen frequency of the visual stimulus appearing on the screen (the SSVEP). Thus, it should be taken care that the frequency of the stimulus as presented on the screen remains constant. Flickering stimuli imprecisely (i.e. varying in frequency) will hamper a correct determination of the user's intent.

This paper illustrates a method to get accurate timings from a PC running Microsoft Windows and how to present visual stimuli on the screen consistently and precisely in time. These presented methods are then used and tested in the development of complete presentation software package (named PTT) that enables students and researchers to quickly design and conduct, among other EEG-based BCI paradigms, simple P300 and SSVEP EEG experiments.

2. RELATED WORK

2.1 NEUROBEHAVIORAL PRESENTATION®

Neurobehavioral Systems' Presentation® is a commercial software, which allows for a precise and powerful stimulus delivery environment for neuroscience experiments [8]. The software itself deliveres with so many features, i.e., stimuli can be 2D visual, 3D visual, auditory, and compressed video.

However, operating the software may be difficult to novice users. To implement an experiment within this software, two custom languages are used (no GUI). One is Scenario Description Language (SDL), which is used to specify stimuli, sequences of stimuli, and all the associated properties. The other one is Presentation Control Language (PCL), which is used to implement custom control of scenarios (an experiment implemented in this software will consist of one or more scenarios). Moreover, hardware configuration is required to be spcified prior to running the experiment (e.g. specifying which display adapter to use for stimulus presentation).

Neurobehavioral Systems' Presentation® may therefore be unsuited for some users (mostly students) who wish to conduct simple EEG experiments, but they will have to spend quite a lot of time learning how this software works and how to write SDL and PCL.

2.2 PSYCHTOOLBOX

Psychtoolbox (Psychophysics Toolbox) is a free software package that is an add-on to Matlab [9]. It supports good timing accuracy in stimulus presentation.

However, Psychtoolbox requires users to have knowledge in programming in order to create EEG experiments.

To enable people with no programming skills to create visual stimuli suitable for EEG experiments, the easy-to-use PTT software was developed so that the user can design his experiment using a GUI.

2.3 WINDOWS TIMING PRECISION

De Clercq et al. [10] states that because Microsoft Windows is not a real-time operating system, it is still unclear whether reaction time programs under these operating systems can be accurate to a millisecond. In 1999, Myors [11] showed that the Microsoft Windows operating system is unable to support precise timing for psychological experimental research. Based on Myors's result, other research studies proposed the use of extra hardware to get accurate timing [10][12]. However, Chambers [13] reexamined this issue in 2003 and revealed some possibilities of getting accurate timing by using timing functions that were newly introduced at the time by Microsoft. Those functions are *QueryPerformanceCounter,* which in conjunction with *QueryPerformanceFrequency* can be used to get a high-resolution system timer with a theoretical resolution of 0,8381 *μsec*. The combination of the two functions will provide the possibility to get a very high resolution of the timing of a critical event happening on the machine.

3. METHODS

3.1 GUI OF PTT SOFTWARE

We have developed a software package, PTT, that aims to provide a user-friendly application to design and run experiments. The software GUI is shown in Fig.1.

For the designing of an experiment, the software has the ability to draw shapes, write text and load the bitmap images onto the screen. Stimulus' properties like size, position, color, font, timing, and flickering frequency can be easily adjusted via the GUI. Stimuli can be grouped into sequences, where each sequence has its associated properties like name, code and timing. Sequences can be either deep-copied or cloned. When a sequence is modified, the changes will propagate to its clones. Yet, all changes made to a deep-copied sequence will not propagate to its original one at all. For the experimental stimulation, all defined sequences will play one by one accordingly.

3.2 TIMING FUNCTIONS IN WINDOWS

Two system functions, named QueryPerformanceCounter and QueryPerformanceFrequenc, are used to get accurate timing from machine. QueryPerformanceFrequency returns the frequency value of computer's system clock, and QueryPerformanceCounter returns the number of counter ticks of this clock since the time at which system starts to boot. For example, the following code returns the time the system needed to execute a program section:

```
LARGE_INTEGER f;

QueryPerformanceFrequency(&f);

freq = f.QuadPart;

LARGE_INTEGER time;

QueryPerformanceCounter(&time);

counter = time.QuadPart;

  <Program section>

QueryPerformanceCounter(&time);

timeDuration = (time.QuadPart - counter)/freq;
```

First, QueryPerformanceCounter sets the variable time to the time at which the program section begins. At the end of program section, QueryPerformanceCounter is called again, and set the variable time to the time at which the program section ends. Calculating the differences between these two values and dividing them by the frequency value obtained from calling QueryPerformanceFrequency can achieve a very precise timing of a specified period.

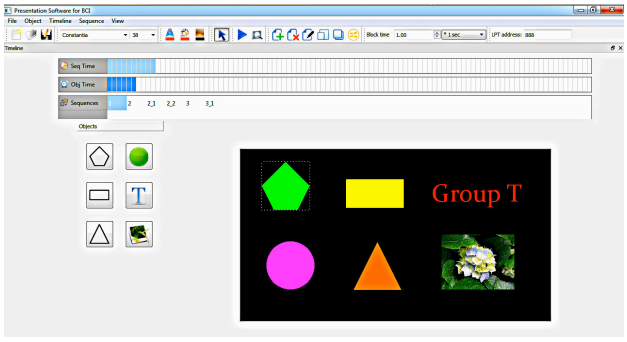

Fig. 1. GUI of PTT software

3.3 DRAWING STIMULI USING OPENGL

The reason why OpenGL is used for this context is that OpenGL provides opportunities to have full control of the hardware, and this brings the advantage of drawing stimuli with specified frequencies precisely on the screen.

The rendering context used in our method is a double-buffered context. In double buffering, while the contents of one buffer are being displayed, drawing takes place in the other buffer. When a glXSwapBuffers call is made, the roles of the two buffers are reversed. This means that if buffers are swapped every frame, there is one frame period (e.g.,16.67ms at 60-Hz screen refresh rate) for drawing into a buffer.

Two important OpenGL settings named SwapInterval and AutoSwapBuffer are used. With AutoSwapBuffer set to 0 and the SwapInterval set to 1, the glFinish function, which is supposed to return right after the data is written to the video buffer, will instead return after the contents of current buffer are actually drawn on screen (simply means waiting until the end of frame period).

Stated differently, by setting SwapInterval to 1, OpenGL will only swap the drawing buffer together with the screen refreshing instead of swapping the buffer as fast as possible, and setting AutoSwapBuffer to 0, OpenGL will swap the drawing buffer only when it is called instead of swapping it automatically. This way, the program has more precise control on the timing of screen events.

3.4 PRECISE STIMULUS PRESENTATION

Most LCD screens at this time have a refresh rate around 60 Hz., which means that after every 16.67ms (1/60Hz) the screen will be redrawn. To present a stimulus with a specified frequency, the idea is to alter the intensity of the stimulus (0 or 1) at every frame.

When displaying a SSVEP stimulus that flickers at an assigned frequency, the idea is to form a block wave representing desired intensity corresponding to each stimulus. For optimal performance, this block wave should be an integer division of the screen's refresh rate so that each frame has a unique intensity value. For every frame period ($t_{refresh} = 1/f_{refresh} = 1/60 = 16.67$ms), stimulus's intensity value should occur at the time instants $k/f_{refresh}$ ($k = 1, 2, \ldots n$).

In order to calculate the intensity values for the stimuli, the time at which the screen starts a new frame should be known. One way to get that value is using the OpenGL graphics library to take control over the hardware. As illustrated in Fig. 2, the red line indicates the time when some drawing codes are needed to determine the stimulus intensity, denoted $t_{current}$. With SwapInterval set to 1, glFinish() will return only at the edge of frame period. By calling QueryPerformanceCounter right after glFinish, we keep track of the previous screen refresh times $t^0_{refresh} \ldots t^n_{refresh}$ and use these values to estimate the screen refresh rate f_{refresh} and finally extrapolate the next refresh time $t^{n+1}_{refresh}$:

$$f_{refresh} = \frac{n}{\sum_{i=1}^{n} (t^i_{refresh} - t^{i-1}_{refresh})} \quad (1)$$

$$t^{n+1}_{refresh} = t^n_{refresh} + \frac{\left(\left\lfloor \left(t_{current} - t^n_{refresh} \right) . f_{refresh} \right\rfloor + 1 \right)}{f_{refresh}} \quad (2)$$

At the time of drawing, the block wave is sampled at time $t^{n+1}_{refresh}$. This way, the flicker frequencies can be controlled precisely, and accurate timings are finally obtained for EEG experiments.

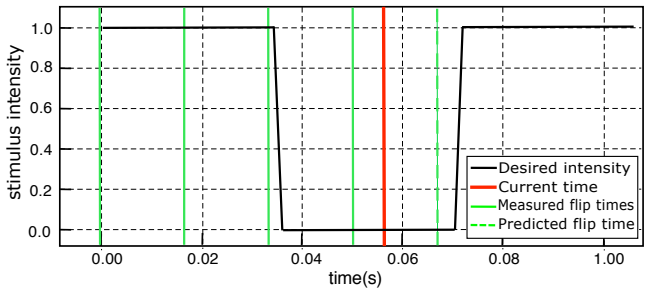

Fig. 2. The value of the stimulus intensity varies in 0 and 1 (black line). On the time axis the measured refresh periods are indicated by green lines. The red line indicates the moment at which the next intensity value is calculated. The estimated moment at which the stimulus is redrawn on the screen is indicated by the green.

4. RESULTS

In order to check how accurate stimuli are presented on screen, some experiments using PTT program are conducted.

The first experiment is to test a SSVEP paradigm, which aims at illustrating how precise the flickering frequency is. A stimulus is drawn on the screen with the assigned frequency of 15Hz, displaying for one minute. While the stimulus is being shown, a light sensor is used to measure the luminosity of the screen. Fig.3 shows the frequency amplitude spectrum of this measurement. The flickering frequency is accurately 15Hz as expected.

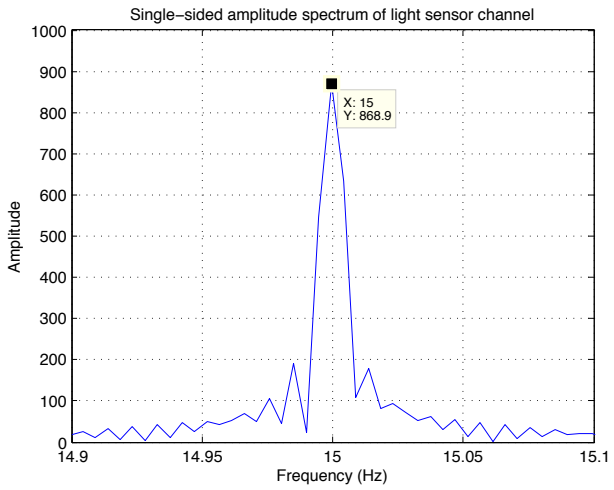

Fig. 3. Spectrum of stimulus's flickering frequency recorded using light sensor.

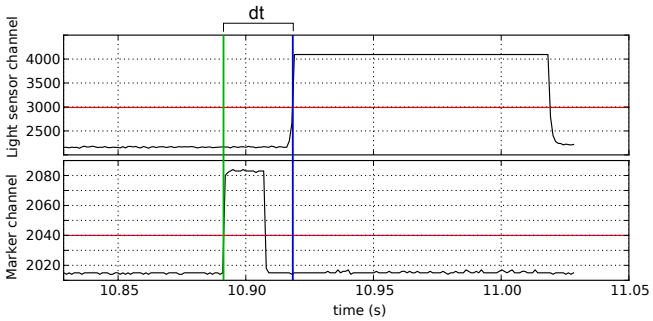

Fig. 4. A measurement of time differences between the marker time and light sensor time. The onset of the events is measured as the moment the upwards-flank of the signal crosses a predefined threshold.

The second experiment is for ERP (Event Related Potentials) like the P300 paradigm. The experiment is conducted with the purpose of getting the time lapse between the presentation of the stimulus on the screen and the time that the synchronization marker is sent over the parallel port to the EEG hardware. Some time lapses are unavoidable because program instructions are executed one by one. First, we wait for the screen to refresh and then we send the marker to the parallel port. This fixed time lapse is not problematic when averaging ERPs since each ERP will have the same offset. We, therefore, strive for the lowest variation of the time lapse. In the experiment, 120 sequences are created. Each sequence contains one stimulus that is presented on screen for 200ms followed by 300ms of blank screen. At the onset of the sequence, a marker value of 255 is sent out through the parallel port. In the meantime, a light sensor is used to detect the time at which the stimulus really appears on screen, which is recorded simultaneously with the voltage on the pins of the parallel port. As indicated in Fig. 4, the time lapse between these two channels can be measured as *dt*. As a result from this experiment, for 54 trials, *dt* is between 26–27ms, and for the remaining 66 trials, *dt* is between 27–28ms. So the timing is accurate up to 2ms, which is good enough for simple EEG experiments.

4. CONCLUSION

The timing accuracy of presenting stimuli has always been a major concern in most psychophysical experiments.

In this paper, the earlier research regarding this timing accuracy is reviewed. Also some existing software is presented, which attempts to solve this accurate timing problem. Importantly, we presented a new method for accurately presenting stimulus with a specified frequency.

The PTT software package has been developed to enable people with no programming skills to quickly design and conduct EEG experiments. Even more, the PTT software achieves accurate and precise timing of stimulus presentation. Measurements with a light sensor indicate the PTT software delivers a satisfactory result.

References

[1] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller and T. M. Vaughan: “Brain-computer interfaces for communication and control”, *Clinical Neurophysioloy,* Vol.113, pp. 767-791, 2002.

[2] L. A. Farwell, E. Donchin: “Talking off the top of your head: toward a mental prothesis utilizing event-related brain potentials”, *Electroenceph clin Neurophysiol*, Vol. 70, pp. 510-523, 1988.

[3] E. Donchin, K.M. Spencer, R. Wijesinghe, (2000). The mental prothesis: assessing the speed of a P300-based brain-computer interface, IEEE Trans. Rehabil. Eng.,8,174-179.

[4] D. Regan: “Human Brain Electrophysiology: Evoked Potentials and Evoked Magnetic Fields in Science and Medicine”, *New York Elsevier*, 1989.

[5] F. Beverina, G. Palmas, S.Silvoni, F. Piccione: “User adaptive BCIS: SSVEP and P300 based interfaces”, PsychNology, Vol 1, pp. 331-354, 2003.

[6] S. T. Morgan, J.C. Hansen and S. A. Hillyard: “Selective attention to stimulus location modulates the steady-state visual evoked potential”, *Neurobiology*, Vol. 93, pp. 4770-4774, 1996.

[7] M.M. Muller, S.A. Hillyard, (1997), Effects of spatial selective attention on the steadystate visual evoked potential in the 20-28 hz range, Cognitive Brain Research, 6,249-261.

[8] Neurobehavioral Systems. Presentation® software. [Online], Available: http://www.neurobs.com/menu_presentation/menu_features/features_overview, March 2012.

[9] D. H. Brainard, D. G. Pelli, M. Kleiner. Psychophysics Toolbox. [Online], Available: http://www.psychtoolbox.org/HomePage, March 2012.

[10] A. D. Clercq, G. Crombez, A. Buysse, and H. Roeyers: “A simple and sensitive method to measure timing accuracy”, *Behavior Research Methods,* Vol. 35, No. 1 pp. 767-791, 2003.

[11] B. Myors: “Timing accuracy of PC programs running under DOS and Windows”, *Behavior Research Methods,* Vol. 31, pp. 322-328, 1999.

[12] R. Richard, Plant, N. Hammond, and T. Whitehouse: “Toward an Experimental Timing Standards Lab: Benchmarking precision in the real world”, *Behavior Research Methods*, Vol.34, No. 1, pp. 218-226, 2002.

[13] C. D. Chambers: “Timing accuracy under Microsoft Windows revealed through external chronometry*”, Behavior Research Methods,* Vol.35, No.1, pp. 96-108, 2003.