Towards Accurate Browser-based SSVEP Stimuli Generation

Alison Camilleri[1][a], Chris Porter[1][b] and Tracey Camilleri[2][c]
[1]*Department of Computer Information Systems, University of Malta, Msida, Malta*
[2]*Department of Systems and Control Engineering, University of Malta, Msida, Malta*

Keywords: Brain Computer Interface (BCI), Steady State Visually Evoked Potential (SSVEP), Stimuli Generation, Web-based SSVEP BCI, Human-Computer Interaction (HCI).

Abstract: Breakthroughs in Brain-Computer Interfacing (BCI) have positively impacted the lives of individuals who suffer from highly-restrictive physical disabilities. BCIs based on Steady State Visually Evoked Potentials (SSVEPs) rely on a neuronal response which takes place in the brain's visual cortex whenever a person focuses visual attention onto a flickering stimulus. Specialized hardware and software tools exist for stimuli generation, however little to no empirical evidence exists on the applicability of standard web technologies for producing accurate and stable stimuli, for use in BCI applications. With the aim of informing efforts for the development of lightweight, portable and low-cost browser-based BCIs, this paper produces initial evidence on the performance attained by widely-adopted web technologies, namely CSS and WebGL. Results demonstrate that for the square wave approximation method, CSS and WebGL are able to effectively render stable and accurate stimuli on both Google Chrome and Mozilla Firefox.

1 INTRODUCTION

Alternative and Augmentative Communication (AAC) tools are introduced to assist people who are unable to communicate due to reduced communication abilities, generally caused by a variety of motor or neurological disorders. AAC refers to any form of aid that can be used to reduce communication barriers, from simple boards showing symbols of day-to-day objects or actions, to rudimentary hardware, such as a wand to point to or select actions, as well as sophisticated eye tracking systems or brain-computer interfaces (BCIs) (Glennen and DeCoste, 1997). BCIs typically rely on electroencephalography (EEG) sensors to capture brain activity, making them beneficial for individuals with reduced physical abilities. However, a number of challenges still stand, particularly concerning accuracy, cost, adoption, portability and convenience outside a lab setting (Maggi et al., 2008).

BCIs based on Steady State Visually Evoked Potentials (SSVEPs) are considered to be robust and require minimal user training to operate. When interacting with an SSVEP-based BCI, electrodes positioned over the occipital region of the brain register frequencies equal to that of a flickering stimulus the user would be attending to. SSVEP detection algorithms are able to determine which stimulus the user is focusing on, and to utilize this information to execute a corresponding control function, for example, to switch between TV channels. In SSVEP-based BCIs, the accuracy and stability of generated stimuli reflects on the quality and in turn the usability of the interface (Cecotti et al., 2010).

Literature shows that most SSVEP stimuli generators are built using desktop-based technologies, such as C++ (Wang et al., 2010), Psychtoolbox with Matlab (Abbasi et al., 2015), OpenGL (Boyd and Chen, 2012), C# (Lalor et al., 2005), and Java (Hasan et al., 2015). This kind of software possesses inherent performance guarantees, as it is able to access native resources, without being bound within restrictive runtimes (e.g. browser engine). On the other hand, minimal research has been carried out to determine the applicability of web technologies for building accurate, lightweight, and portable SSVEP stimulation applications. A few studies have focused on web browser interaction, but for the most part, these make use of desktop-based software for stimuli generation, as in the case of 'WeBB' (Yehia et al., 2017). Recent efforts have led to some experimentation with

---

[a] https://orcid.org/0000-0002-3893-6985
[b] https://orcid.org/0000-0003-3813-2941
[c] https://orcid.org/0000-0002-4908-1863

Camilleri, A., Porter, C. and Camilleri, T.
Towards Accurate Browser-based SSVEP Stimuli Generation.
DOI: 10.5220/0010159400740083
In *Proceedings of the 4th International Conference on Computer-Human Interaction Research and Applications (CHIRA 2020)*, pages 74-83
ISBN: 978-989-758-480-0

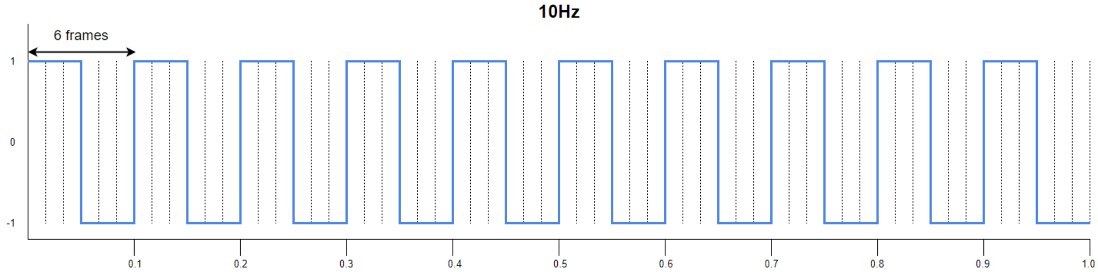

Figure 1: 10Hz stimulus presentation using a constant-period approach.

browser-based stimuli generators, including the use of Graphics Interchange Format (GIF) files (Saboor et al., 2018) and Cascading Style Sheets (CSS) (Saboor et al., 2019) to build online spellers. (Rezazadeh and Sheikhani, 2017) have also succeeded in building an in-browser stimuli interface, for navigating a virtual home in V. Realm Builder.

Notwithstanding this, to the best of our knowledge, little to no empirical evidence exists on the extent to which different web technologies, as well as underlying browser engines, are able to generate accurate and stable stimuli for efficient and usable SSVEP-based interfaces. Furthermore, the viability of web technologies for presenting on-screen stimuli using approximation methods, such as square waves with varied cycle lengths (Wang et al., 2010) has not yet been explored.

This paper explores this research gap, by investigating the adequacy of standard and widely-available web technologies, for generating accurate and stable SSVEP stimuli, which may be adopted in web-based BCI applications.

2 BACKGROUND

The primary purpose of a BCI system is to extract reliable features from brain signals that can be used to classify users' intentions accordingly. A BCI fulfills its purpose in four stages, namely: signal acquisition, feature extraction, feature translation and device control or output. The procedure starts off by recording the electrical activity produced by the brain, typically done by means of non-invasive scalp electrodes. The recorded brain signals are very small waves, therefore they must be amplified and digitized before any further processing can take place. Additionally, the signal of interest and the background signals must be discriminated and separated from one another so that only the relevant signal is analysed. After this process has been successfully completed, signal characteristics are translated into meaningful commands which in turn perform some function, such as controlling an output device (Shih et al., 2012).

This paper considers the use of SSVEPs, which are resonance phenomena that occur whenever a person focuses attention on a visual stimulus that flickers at a frequency greater than 5Hz. In such a scenario, scalp electrodes positioned over the occipital region of the brain, register frequencies equal to that of the flashing stimulus, as well as a number of harmonic/subharmonic frequencies (Zhu et al., 2010). LCD and CRT monitors are typically used to render visual stimuli which are modulated at specific frequencies.

The success rate of an SSVEP-based BCI is highly-dependent upon the choice of visual stimulator (Wang et al., 2008). Having explored various stimuli-rendering methods, (Cecotti et al., 2010) propound that the stability of generated stimuli is crucial for producing effective SSVEP responses. Thus, based on these assertions, technologies which render stimuli at stable frequencies should be preferred, as these are expected to produce better, stronger responses (Cecotti et al., 2010).

As discussed in (Teng et al., 2011), flickering stimuli are typically presented by means of square waves having a 50% duty cycle, with a constant period throughout (Figure 1 shows such an example for a 10Hz stimulus). Using this technique, only frequencies which are integer divisors of the screen refresh rate can be presented in a stable manner. For this reason, complex applications, for which a substantial number of targets is required, such as a phone-dialling program or a speller, cannot present a stimulus for each possible input. These limitations have hindered the development of practical BCI applications, resulting in decreased Information Transfer Rates (ITRs), as users are required to perform a greater number of steps in order to select a desired target (Nakanishi et al., 2014a).

Alternative methods for stimuli presentation exist, including the use of triangular, sine, (Teng et al., 2011) and square waves with varying periods and duty cycles (Wang et al., 2010). Literature indicates that

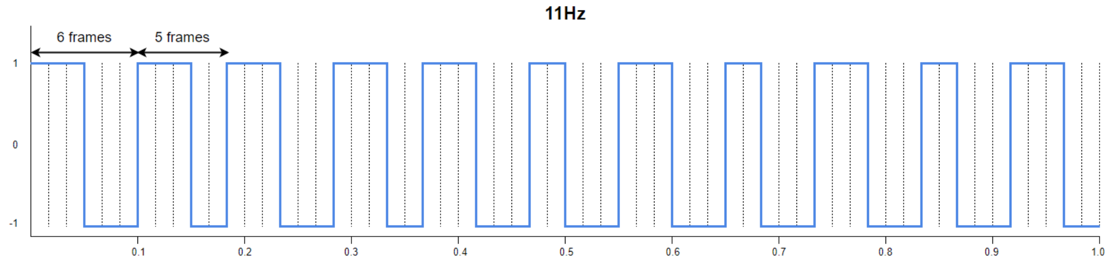

Figure 2: 11Hz stimulus presentation using the square wave approximation approach.

square waves are able to evoke stronger SSVEP responses, when compared to sine or triangular waves (Teng et al., 2011; Chen et al., 2019). Using this technique, the presentation rate is estimated, such that periods consist of a varying number of frames in each cycle (Wang et al., 2010). Stimulus signals may be computed using the following equation, where f represents the stimulus frequency, i is the frame index, and *square()* generates a square waveform with a 50% duty cycle:

$$s(f,i) = square[2\pi f(i/RefreshRate)] \qquad (1)$$

In this manner, stimuli frequencies such as 11Hz, may be presented on a 60Hz monitor, by using periods of different lengths for each cycle, each consisting of five or six frames, which correspond to 12Hz and 10Hz respectively (Wang et al., 2010) (refer to Figure 2). Thus, the approximation of a specific frequency, considers two neighbouring frequencies, derived from the constant-period approach (Nakanishi et al., 2014a). Using this technique, any frequency lower than half the monitor's refresh rate can be approximated (Nakanishi et al., 2014a), thus drastically increasing the number of concurrent stimuli which can be displayed on-screen.

In addition to approximation methods, phase-coding may also be adopted, so that stimuli are discriminated by means of differing phase shifts. (Lee et al., 2010) demonstrated this technique through the development of a multi-target, phase-coded BCI, which presented eight 31.25Hz stimuli using different phases. Therefore, through the application of this technique, stimuli with identical frequencies may be utilized for building SSVEP stimulation interfaces, as long as these are coded to have dissimilar phases (Nakanishi et al., 2014a).

3 AIMS AND OBJECTIVES

As stated by (Saboor et al., 2019), the greatest challenge stands in determining the most effective technology for in-browser rendering of stimuli. The applicability of web technologies, such as Web Graphics Library (WebGL), Flash or Synchronized Multimedia Integration Language (SMIL) is not yet known, thus indicating a gap in research (Saboor et al., 2019). This work will not make use of SMIL or Flash, due to lack of support and adoption by major browser vendors. The de facto technologies used for web graphics and animation rendering will be considered, namely CSS and WebGL.

Furthermore, to the best of our knowledge, no research has been performed to determine the efficacy of approximation techniques within a web browser context. Provided that these are attainable, a greater number of concurrent stimuli could be displayed, allowing for the implementation of more complex web-based SSVEP interfaces.

The aim of this paper is primarily to provide empirical evidence, as to whether SSVEP stimuli can be accurately generated within a web environment, considering standard web technologies (i.e. CSS and WebGL), over two major browsers (Google Chrome and Mozilla Firefox, which are available across all platforms). Moreover, this study seeks to determine whether stimuli produced via square wave approximations, using standard web technologies, are applicable for generating robust SSVEP responses.

This data will take us one step further towards building fully fledged, accurate, usable and highly portable SSVEP-based web browsing interfaces.

The rest of the paper is divided as follows: Section 4 presents the design of the experiment, highlighting the four developed web-based SSVEP stimulation libraries and the main experimental setup. Section 5 discusses the implementation. The results are then presented in Section 6, while the discussion and conclusions are presented in Section 7.

4 EXPERIMENT DESIGN

4.1 Overview

For the purpose of this study, constant-period and square wave approximations were used to build a total of four SSVEP stimulation applications. Thus, for both CSS and WebGL, two stimulators were developed, such that constant-period and approximation versions were implemented for each selected web technology. CSS is a client-side language used to enhance page content presentation. Browser vendors build support for CSS based on standardised published specifications. As opposed to CSS, which operates on the document object model (DOM), WebGL is specifically designed for advanced 2D and 3D graphics rendering on the web, using canvas elements, which offer enhanced performance for graphics-intensive web applications. WebGL is also widely supported by major browsers, including modern versions of Google Chrome, Mozilla Firefox, Safari, Microsoft Edge and Internet Explorer.

The web-based SSVEP stimulation libraries were tested on a single high-spec machine, using both Google Chrome and Mozilla Firefox, in order to observe how their stimuli-rendering performance varies over time. Throughout the different experimental runs, data logs for overall CPU, GPU and memory consumption were recorded. A UNI-T UT372 tachometer was also used to externally measure the on-screen frequency as produced by the stimuli generators (refer to Figure 3 for experimental setup). This was set up in a fixed position with a sampling rate of 255 readings per second. Tachometer data was transferred over USB to a host computer, making it possible to determine the accuracy and stability of the generated stimuli, by means of the selected technologies, when these were run on different browser engines.

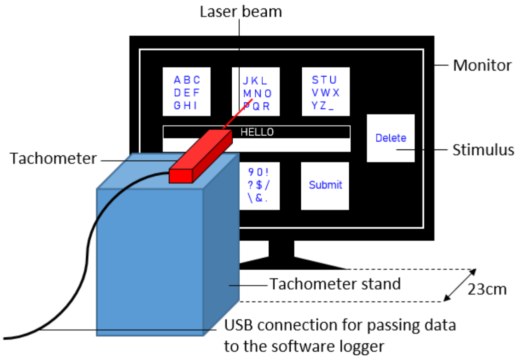

Figure 3: Experimental setup.

4.2 Experimental Procedure

Tests were conducted on a Windows 10 Pro (64-bit) machine, having an Nvidia GTX1080 GPU, Intel Core i7-9700 CPU and 32GB RAM. Stimuli were presented using an ASUS monitor (version VS228), running at a 60Hz refresh rate (1920 x 1080 pixels). For both CSS and WebGL, two 7-target spellers were developed, such that each technology was used for implementing two stimuli generators, using the constant-period and approximation methods respectively. In this way, stimuli frequencies of 6Hz, 7.5Hz, 9Hz, 10.5Hz, 12Hz, 13.5Hz and 15Hz could be presented simultaneously on-screen. For the CSS and WebGL approximation techniques, a total of three test runs were conducted for five selected frequencies, namely 7.5Hz, 9Hz, 10.5Hz, 12Hz and 13.5Hz. Since constant-period methods are unable to generate frequencies which are not integer divisors of the screen refresh rate, the 9Hz, 10.5Hz and 13.5Hz frequencies were solely measured for the approximation approach, described by means of Equation 1. Tachometer readings for the CSS and WebGL implementations were recorded on both Google Chrome (Version 78.0.3904.97 (Official build)(64-bit)) and Mozilla Firefox (Version 70.0.1 (64-bit)). Both web browsers were set up to make use of Hardware Acceleration, allowing for the majority of graphical intensive tasks to be handled by the GPU.

Each target stimulus frequency was measured for four minutes by means of an external tachometer (see Figure 3). This process was repeated three times, such that for each stimuli-rendering method (i.e. constant-period and square wave approximations), twelve minutes' worth of data were gathered for every stimulus frequency, generated via a specific web technology and browser combination. For all test runs, a constant distance of 23cm was maintained between the monitor and tachometer device.

The strength of generated SSVEP responses greatly depends upon the stability and accuracy of rendered stimuli. Hence, for each frequency data set, tachometer values were analyzed in terms of standard deviation (denotes stimulus stability), as well as discrepancies between mean measured frequencies and target frequencies (denotes stimulus accuracy). Furthermore, the Independent Samples t-test was applied to identify any statistically significant differences between two rendering scenarios (for instance, a 7.5Hz CSS stimulus running on Chrome vs. Firefox, or a 7.5Hz stimulus running on Chrome generated via CSS vs. WebGL), in terms of mean measured frequencies. Through this analysis, recommendations could be made regarding technology-browser combinations which yield higher levels of performance. Finally, Cohen's *d* formula was computed for those instances where significant discrepancies were produced by the t-test, in order to appropriately quantify the magnitude of these dissimilarities.

5 IMPLEMENTATION

To fulfill the aims of this study, and as discussed in earlier sections, a total of four SSVEP stimulation libraries were developed, using CSS and WebGL. For each technology, two stimuli-rendering approaches were considered, namely the constant-period and square wave approximation methods. The implemented libraries allow for on-page element animation, making them suitable for the development of flexible, modular and customisable SSVEP interfaces.

Prior to stimulus presentation, each stimulator exploits JavaScript's `requestAnimationFrame()` function to calculate an initial value for the screen refresh rate. In this manner, frame sequence computations are based on the actual screen refresh rate, thus ensuring the stability of generated flickers.

For the CSS and WebGL constant-period approaches, visual flickers were realized by alternating between dark-coloured and light-coloured frames at a constant frequency (Nakanishi et al., 2014b). Thus, taking into consideration a 60Hz monitor, a stimulus frequency of 10Hz would be represented as follows: "111000111000111000111000111...", whereby 1 and 0 represent a dark-coloured and light-coloured frame respectively. As for the square wave approximation approach, this made use of Equation 1, to generate frame sequences for each SSVEP stimulus. In this case, stimuli frequencies, such as 11Hz, may be accurately represented on a 60Hz monitor, by varying the number of frames in a stimulation cycle. Therefore, the square wave approximation technique, would represent an 11Hz stimulus, by alternating between 5 and 6 frame periods as follows: "1110001110011100011100111...".

CSS implementations made use of animation keyframes for the realization of stimuli flickers. For the constant-period method, the number of frames in each stimulation cycle was constant throughout, thus by calculating the time difference between two consecutive frames, animation duration for stimulus cycles could be set, and used to alter stimulus state at a regular interval.

For the square wave approximation approach, stimuli flickers are also produced by means of CSS keyframes, which are dynamically-generated via JavaScript, prior to the rendering of stimuli. Keyframes are set up to alter stimulus opacity, based on values produced by Equation 1. In order to set up the keyframe, square wave sequences are pre-computed for the required number of frames. For our implementation, the 'required number of frames' value is determined based on the frame index, at which the square wave cycle repeats itself.

In order to improve rendering efficiency, all CSS keyframes were set up to make changes to elements' 'opacity' property, so that stimulus presentation would involve less rendering stages. According to (Lewis, n.d.), the most performant version of the pixel pipeline eliminates the need for constant repaints, and only requires compositing changes. By promoting a specific element to its own layer, the 'opacity' property can be handled by the compositor alone, thus reducing the number of steps involved in rendering visual flickers.

The WebGL stimulation implementations make use of a single, large canvas for presenting on-screen SSVEP stimuli. Since browsers impose a limit on the number of concurrent WebGL contexts, the use of a canvas element for each stimulus was not considered to be a viable option. By using a canvas which covers the entire window, placeholder `<div>` elements can be used to set stimuli locations. Through the use of `element.getBoundingClientRect()`, the viewport is set to cover the `<div>`'s location, so that stimulus flickers may be generated within the specified area. Prior to the generation of on-screen stimuli, two textures are rendered for each stimulus; a dark-coloured texture, and a light-coloured texture. During stimulus presentation, `WebGLRenderingContext.bindTexture()` is invoked to display dark/light textures at specific time intervals.

For WebGL's constant-period approach, the frame sequence was computed for a single period, and stored within an array, so that it could be repeatedly used for stimulus presentation. In this way, for every `requestAnimationFrame()` invocation, a specific frame value is read out of the array, to display the appropriate texture. For instance, a frame value of 0 would indicate that a black texture is to be shown, while a frame value of 1 would signify white texture presentation.

A similar approach was also adopted for WebGL's approximation technique, however, as discussed previously, square wave sequence values were computed via Equation 1, for the required number of frames. The resultant values were also stored within an array, and looped over continuously to display appropriate stimuli flickers.

Table 1: Table showing results for the square wave approximation method, including the average bias and standard deviation (calculated for 3 test runs) for each technology, browser, and stimulus frequency combination.

Technology	Web Browser	Actual Freq. (Hz)	Mean Measured Freq. (Hz)	Bias	Standard Deviation
CSS	Chrome	7.5	7.49982786	0.00017214	0.000357508
CSS	Chrome	9	8.999826161	0.000173839	0.000033826
CSS	Chrome	10.5	10.510888396	0.010888396	0.016871103
CSS	Chrome	12	11.999742734	0.000257266	0.00086882
CSS	Chrome	13.5	13.499745745	0.000254255	0.021762828
Mean				**0.002349179**	**0.007978817**
WebGL	Chrome	7.5	7.499837479	0.000162521	0.000752463
WebGL	Chrome	9	8.999831297	0.000168703	0.0000183087
WebGL	Chrome	10.5	10.511127822	0.011127822	0.016984468
WebGL	Chrome	12	11.999758494	0.000241506	0.000930272
WebGL	Chrome	13.5	13.50000045	0.000000445	0.022015793
Mean				**0.002340199**	**0.008140261**
CSS	Firefox	7.5	7.500371427	0.000371427	0.003879366
CSS	Firefox	9	9.000257754	0.000257754	0.003808133
CSS	Firefox	10.5	10.510751762	0.010751762	0.017077129
CSS	Firefox	12	12.000036605	0.000036605	0.003783792
CSS	Firefox	13.5	13.500389542	0.000389542	0.023047437
Mean				**0.002361418**	**0.010319171**
WebGL	Firefox	7.5	7.499832556	0.000167444	0.0000197006
WebGL	Firefox	9	8.999830261	0.000169739	0.0000224204
WebGL	Firefox	10.5	10.511080384	0.011080384	0.017025472
WebGL	Firefox	12	11.998369124	0.001630876	0.02129115
WebGL	Firefox	13.5	13.499634118	0.000365882	0.02178422
Mean				**0.002682865**	**0.012028593**

6 RESULTS

6.1 Stimuli Accuracy and Stability

In the context of an SSVEP-based BCI, stimuli stability and accuracy are crucial for the generation of effective SSVEP responses. Hence, for the purpose of this analysis, technology and browser combinations are considered to be highly performant, if they render stimuli at frequencies which are very close to their expected values, with minimal variance over time.

Thus, to determine the applicability of web technologies for efficacious generation of SSVEP responses, tachometer data was initially analyzed in terms of (a) dispersion of values (denotes stimulus stability), as well as (b) discrepancies between mean measured frequencies and target frequencies (denotes stimulus accuracy). For each stimuli-rendering method, technology and browser combination, both mean and standard deviation values were computed, for stimulus frequency measurements captured by the tachometer. This procedure was repeated for frequency data sets pertaining to a single test run, as well as amalgamated data sets, consisting of frequency values for the three separate runs.

Mean values for merged data sets were subsequently used to compute a value for the 'bias', which is defined as the absolute value of the mean measured frequency, subtracted from the target frequency (refer to Equation 2). The bias value indicates the extent to which mean measured frequencies differ from their expected values.

$$Bias = |Target\,Frequency - Mean\,Frequency| \quad (2)$$

For both constant-period and approximation methods, average values for the bias and standard deviation were also calculated, taking into consideration all stimuli frequencies, generated via a specific technology and browser combination (refer to Table 1 for square wave approximation results). These values may be considered as performance indicators, which demonstrate the technology and browser's ability to generate stimuli at stable and accurate frequencies.

Stimuli generated via constant-periods were found to make use of the exact frame sequences as square wave approximations, for presenting stimuli at 7.5Hz and 12Hz (integer divisors of the screen refresh rate). Hence, results in terms of bias and standard deviation did not yield significant discrepancies across these two rendering approaches. For this reason, this paper shall focus on discussing the applicability of distinct web technology and browser permutations, for presenting stable and accurate stimuli via square wave approximations.

From results obtained, it is evident that high performance levels were attained for each target fre-

Table 2: Summary of t-test and Cohen's *d* formula results (calculated for the amalgamation of 3 test runs) for stimuli generated via the square wave approximation method, under the various rendering scenarios.

Stimulus		Comparison	P-Value (T-Test)	Cohen's *d* Value
CSS	7.5Hz	Chrome vs. Firefox	**0**	0.19732
CSS	9Hz	Chrome vs. Firefox	**0**	0.16027
CSS	10.5Hz	Chrome vs. Firefox	0.881	-
CSS	12Hz	Chrome vs. Firefox	**0**	0.10705
CSS	13.5Hz	Chrome vs. Firefox	0.184	-
WebGL	7.5Hz	Chrome vs. Firefox	0.666	-
WebGL	9Hz	Chrome vs. Firefox	**0.016**	0.05062
WebGL	10.5Hz	Chrome vs. Firefox	0.827	-
WebGL	12Hz	Chrome vs. Firefox	**0**	0.09220
WebGL	13.5Hz	Chrome vs. Firefox	0.424	-
Chrome	7.5Hz	CSS vs. WebGL	0.442	-
Chrome	9Hz	CSS vs. WebGL	**0**	0.18884
Chrome	10.5Hz	CSS vs. WebGL	0.317	-
Chrome	12Hz	CSS vs. WebGL	0.386	-
Chrome	13.5Hz	CSS vs. WebGL	0.59	-
Firefox	7.5Hz	CSS vs. WebGL	**0**	0.19644
Firefox	9Hz	CSS vs. WebGL	**0**	0.15875
Firefox	10.5Hz	CSS vs. WebGL	0.35	-
Firefox	12Hz	CSS vs. WebGL	**0**	0.10905
Firefox	13.5Hz	CSS vs. WebGL	0.114	-

quency, technology, and browser combination, as signified by the remarkably small mean values achieved for the bias and standard deviation throughout (refer to Table 1). Results show that the largest inaccuracies/instabilities were noted when the WebGL stimuli generator was run using Firefox, resulting in mean bias and standard deviation values of 0.002682865 and 0.01202859252 respectively. Although this technology and browser combination resulted in the highest level of bias and variance, it is worth noting that such degrees of inaccuracy and instability are very small, and are thus considered to be negligible in the context of an SSVEP BCI system. Therefore, stimuli generated under such rendering conditions should still be able to evoke appropriate SSVEP responses.

6.2 Performance Comparisons

Based on the Central Limit Theorem (CLT), the distribution of sample means approaches a normal distribution as the sample grows in size. Irrespective of whether the source population is normal or skewed, this holds true, provided that the sample size is ≥ 30 and sampling is carried out by replacement (LaMorte, 2016). For a specific stimulus frequency, rendered via a specific technology-browser permutation, tachometer data for the amalgamation of three test runs consisted of $\approx$ 4000 data points. Hence, based on CLT criteria, these data sets are considered to have sufficiently large sample sizes, which indicate that data should follow a normal distribution. As a result, parametric tests, such as the Independent Samples t-test may be applied to make the necessary inferences. Since the t-test assumes homogeneity of variance, p-values produced by the Levene's test were initially computed, and used to reference the appropriate t-test result. SPSS produces two p-values for the t-test statistic, which were quoted based on whether the assumption for equality of variances was violated or not (KentStateUniversity, 2019).

The t-test was executed for distinct technology-browser combinations, each time making use of two frequency data sets, consisting of 12 minutes' worth of tachometer data (pertaining to 3 test runs). The level of significance was taken to be 0.05 for all tests, thus $p<.05$ indicated statistically significant discrepancies between stimuli generation technologies, running on different browsers.

Results produced by the Independent Samples t-test are listed in Table 2, to showcase any statistically significant discrepancies between two distinct rendering scenarios (for instance, a 7.5Hz CSS stimulus running on Chrome vs. Firefox, or a 7.5Hz stimulus running on Chrome generated via CSS vs. WebGL). As evidenced in Table 1, all technology-browser combinations fared relatively well, and were able to produce stimuli which remained close to their target frequencies, with minimal deviation over time. Through the application of the t-test, the degree by which these rendering setups differed from one another could be further understood. This test was considered to be a first pass analysis, for the provision of recommendations, regarding technology and browser setups, which should be favoured when building web-based SSVEP stimuli generators.

For stimuli rendered via square wave approximations, it was observed that when running the CSS stimuli generator on both Chrome and Firefox, a total of three significant discrepancies were noted in stimuli-rendering performance. In this case, bias values were very close to one another, however Firefox showed an overall higher variance (Table 1), which indicates that Chrome might be a better alternative for presenting CSS-generated stimuli. In the case of WebGL, similar observations were made, whereby calculated bias values were very close to one another, yet stimuli generated via Firefox had a greater degree of instability. Thus, results achieved are consistent for both technologies, and favour the use of Chrome over Firefox, for presenting stable and accurate SSVEP stimuli.

When running both the CSS and WebGL stimuli generators on Chrome, results obtained were very similar, both in terms of bias and standard deviation (Table 1). Thus, both technologies are considered to be comparable for effective presentation of SSVEP stimuli via Chrome. However, when running the same stimuli generators on Firefox, three out of five frequencies were found to be significantly different, with a p-value less than 0.05 (Table 2). In this scenario, WebGL had an overall higher bias and standard deviation when compared to CSS, which indicates that when using Firefox, CSS might be a better alternative.

In the context of an SSVEP-based BCI, such minor discrepancies are not considered to be of significance, as all stimuli frequencies remained very close to their expected values, and showed minimal variance over time. This variance, which was less than 0.03 across all cases, shows that these stimuli rendering technologies are adequate, even for complex, multi-target systems, such as the 40-target BCI developed by (Nakanishi et al., 2017), where the frequency difference between stimuli was as low as 0.2Hz.

As stated by (Halsey, 2019), p-values provide limited information about our data, and can thus be misinterpreted. P-values tend to exhibit high sample-to-sample variability, which does not reliably indicate the amount of evidence against a specified null hypothesis. Thus, these values should only be used for providing first pass evidence, about a phenomenon being studied (Halsey, 2019).

As opposed to significance testing, estimation statistics are targeted towards determining the magnitude of an effect and its precision. An effect size may simply be described as the “the degree to which the null hypothesis is false” (Enzmann, 2015). A well-known measure of effect size is Cohen’s *d* formula, which compares the means of two independent groups.

For the Independent Samples t-test, Cohen’s *d* value may be determined by calculating the mean difference between two groups, and then dividing this result by the pooled standard deviation. Equation 3 depicts Cohen’s *d* formula (Rosenthal et al., 1994), whereby M_1 and M_2 refer to the means of the first and second samples, and SD_{pooled} is the pooled standard deviation of the two samples:

$$d = \frac{M_1 - M_2}{SD_{pooled}} \tag{3}$$

Equation 4 is used to compute SD_{pooled}, whereby SD_1 and SD_2 represent the standard deviation for the first and second samples respectively:

$$SD_{pooled} = \sqrt{\frac{{SD_1}^2 - {SD_2}^2}{2}} \tag{4}$$

For the purpose of this study, Cohen’s *d* formula was computed to determine the effect size, for instances where significant discrepancies were noted by the t-test (refer to Table 2). When considering large sample sizes, minor effects, such as minute discrepancies between means, can result in significant statistical differences (Walker, 2008). Thus, by making use of Equation 3, the magnitude of differences between two groups can be appropriately quantified.

Results obtained for Cohen’s *d* value (Table 2) were interpreted based on commonly used benchmarks (Lakens, 2013), whereby *d* values equal to 0.2, 0.5, and 0.8, denote small, medium and large effect sizes respectively. Thus, if discrepancies between two group means are smaller than 0.2 standard deviations, differences are considered to be negligible, even if statistically significant (Walker, 2008).

Based on this, values produced by Cohen’s *d* formula (refer to Table 2) indicate that despite significant differences produced by the t-test, these discrepancies correspond to a small effect size ($d<0.2$). Thus, for all permutations listed in Table 2 (refer to the ‘Stimulus’ and ‘Comparison’ columns), the means for stimuli frequencies generated under the various rendering scenarios are similar, and any discrepancies are considered to be negligible. In practice, results attained demonstrate that irrespective of the selected technology-browser combination, appropriate SSVEP responses may be evoked from web-based stimuli generated via square wave approximation methods.

7 CONCLUSIONS

This study provides evidence-based arguments on the applicability of standard browser-based technologies for SSVEP stimuli generation. Results show that CSS and WebGL may be used to render effective SSVEP stimuli, on both Google Chrome and Mozilla Firefox. This was demonstrated by the consistent stability and accuracy of the generated stimuli.

Furthermore, this study successfully adopted the square wave approximation technique, for presenting stimuli frequencies which are non-integer divisors of the screen refresh rate. Using this method, a greater number of concurrent on-screen stimuli can be displayed, thus allowing for the development of complex BCI applications, such as web-based spellers, with a target for each possible input.

Contrary to state of the art SSVEP stimulation technologies, in-browser stimuli generators guarantee cross-platform portability, while further lowering barriers for BCI-enabled web development. Building a web-based BCI also depends on an ecosystem of technologies, bringing forth various challenges; from efficient and cost-effective user-side setup, to remote signal processing, and finally, real-time browser control.

ACKNOWLEDGEMENTS

We would like to thank Dr Sandro Spina from the Computer Graphics and Visualisation Group, for providing insights, as well as state of the art hardware, on which tests were carried out. We would also like to thank Rosanne Zerafa from the Biomedical Cybernetics lab, for insights provided at the initial stages of this study.

REFERENCES

Abbasi, M. A., Gaume, A., Francis, N., Dreyfus, G., and Vialatte, F.-B. (2015). Fast calibration of a thirteen-command bci by simulating ssveps from trains of transient veps-towards time-domain ssvep bci paradigms. In *Neural Engineering (NER), 2015 7th International IEEE/EMBS Conference on*, pages 186–189. IEEE.

Boyd, J. and Chen, Y. (2012). An open source stimulator for ssvep-based bcis. In *Proceedings of the 50th Annual Southeast Regional Conference*, pages 124–129. ACM.

Cecotti, H., Volosyak, I., and Gräser, A. (2010). Reliable visual stimuli on lcd screens for ssvep based bci. In *Signal Processing Conference, 2010 18th European*, pages 919–923. IEEE.

Chen, X., Wang, Y., Zhang, S., Xu, S., and Gao, X. (2019). Effects of stimulation frequency and stimulation waveform on steady state visual evoked potentials using computer monitor. *Journal of Neural Engineering*.

Enzmann, D. (2015). Notes on effect size measures for the difference of means from two independent groups: the case of cohen'sd and hedges'g. *January*, 12:2015.

Glennen, S. and DeCoste, D. C. (1997). *The handbook of augmentative and alternative communication*. Cengage Learning.

Halsey, L. G. (2019). The reign of the p-value is over: what alternative analyses could we employ to fill the power vacuum? *Biology letters*, 15(5):20190174.

Hasan, M. K., Hossain, T. M., Faisal, A. K., and Ahmad, M. (2015). Effect of diverse stimulations of brain on eeg signal for ssvep based brain computer interface. In *2015 International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*, pages 1–6. IEEE.

KentStateUniversity (2019). SPSS Tutorials: Independent Samples T Test. Retrieved Nov 17, 2019, from https://libguides.library.kent.edu/SPSS/IndependentTTest.

Lakens, D. (2013). Calculating and reporting effect sizes to facilitate cumulative science: a practical primer for t-tests and anovas. *Frontiers in psychology*, 4:863.

Lalor, E. C., Kelly, S. P., Finucane, C., Burke, R., Smith, R., Reilly, R. B., and Mcdarby, G. (2005). Steady-state vep-based brain-computer interface control in an immersive 3d gaming environment. *EURASIP Journal on Advances in Signal Processing*, 2005(19):706906.

LaMorte, W. (2016). Central Limit Theorem. Retrieved Nov 17, 2019, from http://sphweb.bumc.bu.edu/otlt/MPH-Modules/BS/BS704_Probability/BS704_Probability12.html.

Lee, P.-L., Sie, J.-J., Liu, Y.-J., Wu, C.-H., Lee, M.-H., Shu, C.-H., Li, P.-H., Sun, C.-W., and Shyu, K.-K. (2010). An ssvep-actuated brain computer interface using phase-tagged flickering sequences: a cursor system. *Annals of biomedical engineering*, 38(7):2383–2397.

Lewis, P. (n.d.). Stick to Compositor-Only Properties and Manage Layer Count. Retrieved Nov 15, 2019, from https://developers.google.com/web/fundamentals/performance/rendering/stick-to-compositor-only-properties-and-manage-layer-count.

Maggi, L., Parini, S., Perego, P., and Andreoni, G. (2008). *BCI++: an object-oriented BCI prototyping framework*. Citeseer.

Nakanishi, M., Wang, Y., Chen, X., Wang, Y.-T., Gao, X., and Jung, T.-P. (2017). Enhancing detection of ssveps for a high-speed brain speller using task-related component analysis. *IEEE Transactions on Biomedical Engineering*, 65(1):104–112.

Nakanishi, M., Wang, Y., Wang, Y.-T., Mitsukura, Y., and Jung, T.-P. (2014a). Generating visual flickers for eliciting robust steady-state visual evoked potentials at flexible frequencies using monitor refresh rate. *PloS one*, 9(6):e99235.

Nakanishi, M., Wang, Y., Wang, Y.-T., Mitsukura, Y., and Jung, T.-P. (2014b). A high-speed brain speller using steady-state visual evoked potentials. *International journal of neural systems*, 24(06):1450019.

Rezazadeh, Z. and Sheikhani, A. (2017). An ssvep-based brain-computer interface to navigate in a virtual home. In *Proceedings of the 7th International Conference on Bioscience, Biochemistry and Bioinformatics*, pages 22–27. ACM.

Rosenthal, R., Cooper, H., and Hedges, L. (1994). Parametric measures of effect size. *The handbook of research synthesis*, 621:231–244.

Saboor, A., Benda, M., Gembler, F., and Volosyak, I. (2019). Word prediction support model for ssvep-based bci web speller. In *International Work-Conference on Artificial Neural Networks*, pages 430–441. Springer.

Saboor, A., Gembler, F., Benda, M., Stawicki, P., Rezeika, A., Grichnik, R., and Volosyak, I. (2018). A browser-driven ssvep-based bci web speller. In *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 625–630. IEEE.

Shih, J. J., Krusienski, D. J., and Wolpaw, J. R. (2012). Brain-computer interfaces in medicine. In *Mayo Clinic Proceedings*, volume 87, pages 268–279. Elsevier.

Teng, F., Chen, Y., Choong, A. M., Gustafson, S., Reichley, C., Lawhead, P., and Waddell, D. (2011). Square or sine: Finding a waveform with high success rate of eliciting ssvep. *Computational intelligence and neuroscience*, 2011:2.

Walker, I. (2008). Null hypothesis testing and effect sizes. Retrieved Nov 17, 2019, from https://people.bath.ac.uk/pssiw/stats2/page2/page14/page14.html.

Wang, Y., Gao, X., Hong, B., Jia, C., and Gao, S. (2008). Brain-computer interfaces based on visual evoked potentials. *IEEE Engineering in medicine and biology magazine*, 27(5):64–71.

Wang, Y., Jung, T.-p., et al. (2010). Visual stimulus design for high-rate ssvep bci. *Electronics letters*, 46(15):1057–1058.

Yehia, A. G., Eldawlatly, S., and Taher, M. (2017). Webb: A brain-computer interface web browser based on steady-state visual evoked potentials. In *Computer Engineering and Systems (ICCES), 2017 12th International Conference on*, pages 52–57. IEEE.

Zhu, D., Bieger, J., Molina, G. G., and Aarts, R. M. (2010). A survey of stimulation methods used in ssvep-based bcis. *Computational intelligence and neuroscience*, 2010:1.