Unity-based BCI2000 Application Layer: Virtual Reality and the Internet of Things

A Thesis
SUBMITTED TO THE FACULTY OF
UNIVERSITY OF MINNESOTA
BY

Christopher Coogan

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF SCIENCE

Bin He

December 2017

© Christopher Coogan 2017

Acknowledgments

There is no way I could have completed this latest journey of my life without the support of so many people. Firstly, Dr. Bin He for allowing me the academic freedom to explore various aspects of the field, and for keeping me on track as I tried to tackle far too many projects. Also, my parents, John and Donna Coogan, for offering unwavering support throughout each of the paths my life has taken me so far. The He Lab, especially Christopher Cline, Bradley Edelman, and Daniel Suma, were paramount in helping me get to this point. I have learned so much from each of them, and have plenty more to learn. Lastly, I'd like to thank my thesis committee, Dr. Stephen Engle and Dr. Zhi Yang, for taking the time and effort to help guide me through this process.

Table of Contents

List of Figures

Introduction

Brain computer interfaces (BCIs) are devices that monitor neurological activity in the brain and create a control signal based upon those signals to be used by an external application. With the advent of commercial-grade virtual reality (VR) devices, graphical processing units (GPUs), the Internet of Things (IoT), and advances robotics, that external application can easily be a videogame, therapeutic virtual experience, or home devices such as coffee pots and televisions. Users can interact with both the physical and virtual world and to any device with access to the internet, via BCIs. BCI research and devices are commonly used in both healthy and clinical populations. While healthy individuals can utilize BCI for recreational applications, disabled individuals can use BCI for rehabilitation or assistive devices. Those individuals with a motor impairment would particularly find IoT applications useful inside their own home by being able to control their personal devices. Likewise, having BCI technology that benefits the healthy population could generate public interest in the field to advance it further. Great advances in the underlying neurophysiological research has been attained in the past several years and we can now harness that to build complementary applications [1-3].

There are various methods of recording neurophysiological signals, each having their pros and cons. For healthy subjects the most often used neural signal acquisition device would be the electroencephalogram (EEG). The benefits of using such as device is that it is noninvasive, has excellent temporal resolution, and more cost effective. However, EEGs record global populations of dendritic currents which must pass through the skull in order to reach the electrodes. This causes signal attenuation [4], reducing the frequencies that can be recorded to no higher than 30 Hz. Electrocorticography (ECoG) operate very similarly to EEG except that the electrodes are placed directly onto the cortical surface. This increases the overall signal to noise (SNR) ratio and allows higher frequencies to be recorded, but also requires surgical implantation and is therefore only available to the clinical population. Likewise, there are also multiple categories neurological control techniques that can be incorporated into brain computer interfaces such as sensorimotor rhythms (SMR), steady-state visual evoked potentials (SSVEP), and the P300 evoked response, all of which are easily discernable using an EEG and ECoG. SMR signals are elicited through the act of motor imagination (MI). By performing an imagined act such the opening/closing of a hand, dorsiflexion of the foot, or protrusion/retraction of the tongue we can record fluctuations in the sensorimotor cortex. In the case of MI of the hand, an event related desynchronization/synchronization can be seen on the contra/ipsilateral cortex in the frequency band of 7-13 Hz. [4] SSVEP occur naturally in the visual cortex when a subject is attentive to a stimulus presented at a particular frequency, if a 7.5 Hz signal is presented to the user, a 7.5 Hz signal can be recorded from electrodes placed above the occipital lobe. The P300 response can be seen 300ms after the user recognizes the stimulus [5]. When a user seeks an object presented at a relatively low frequency, that ‘oddball’ stimuli can be easily discerned as a neural correlate 300ms after stimulation.

In traditional BCI research users are given a particular task utilizing one, or several, of the neurological control techniques. Several examples of BCI successfully utilizing these signals include controlling drones, robotic arms, virtual avatars, video games, and wheelchairs [6-13]. A critical component of the BCI feedback mechanism is the visual stimuli provided to the user to give them an indication of their progress during the task. It is widely accepted in the literature that the type of feedback presented can alter the performance on a given BCI task. [14, 15] Many advances have been made from both a neuroscientific and an engineering perspective with regards to brain computer interfaces, but little has been accomplished from a human-computer interaction point of view. If BCI performance is task-dependent and the task presented to the user does not motivate or incentivize them to perform well, there is room for improvement [2]. What I aim to create is not only devices and applications that provide therapeutic benefits and improvements in the quality of life, but applications that people are comfortably using long term.

Here, I present a tool that easily allows a user to create and utilize novel applications for use with brain computer interfaces. By combining a game engine for content creation, various application programming interfaces (APIs) for the control of external IoT devices, and virtual/augmented reality libraries for immersive experiences, I have created a tool that will allow the creation of the next generation of easy-to-use BCI applications for both research and personal use.

Materials and Methods

BCI2000

Signals from both EEG and ECoG devices can be recorded, analyzed, and utilized using a popular software package, BCI2000 [16]. BCI2000 allows for the acquisition of neural recordings from a wide-range of 3rd party systems (Neuroscan, BioSemi, etc…), formatting and sending that raw data to pre-built or custom signal processing modules, creating a control signal based upon the signal used and the application to be sent to, and finally, the creation of an application to be presented to the end-user.

BCI2000's operating pipeline can be viewed in Figure 1. To begin running an experiment, the user inputs parameters into the operator layer as a series of instructions. These inputs determine what type of signal acquisition device is to be used, what filtering pipeline, and what application the user will interact with, along with subject-specific parameters such as subject info and experiment time. The source module is responsible for communicating with the acquisition hardware. The signal processing

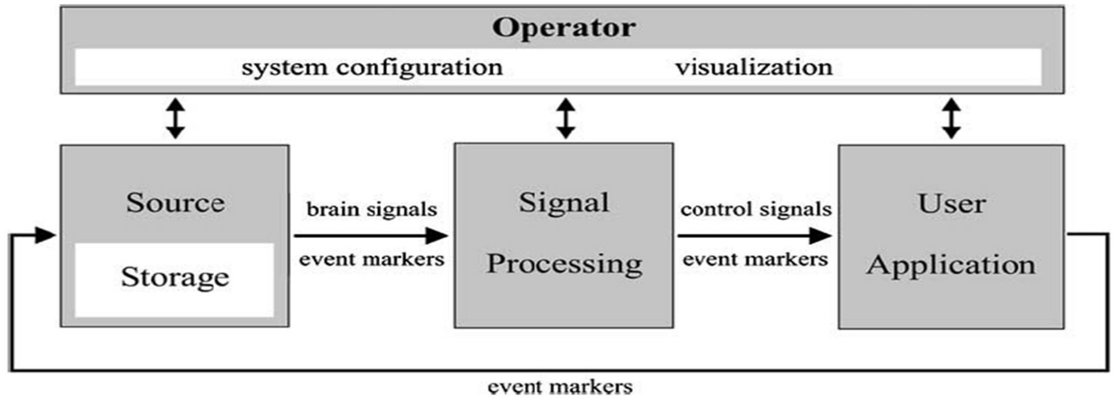

Figure 1: BCI2000 pipeline. Taken from [16]

module receives the output from the source module and creates a control signal to send to the application layer based upon the filter and task.

Native BCI2000 use allows an experimenter to pass commands to the Operator module via command line arguments, shell scripting, batch files, or a graphical user interface (GUI). In the background, a telnet connection is created on a specific IP address and port number, which users utilize by opening a shell window and writing commands directly to the operator module from external applications. These commands determine the network parameters, flags to run multiple BCI2000 instances in parallel, where to route particular signals, and specify variables to trigger during particular tasks in the application layer. Because of BCI2000's modularity of components, supplanting the underlying application layer only requires forwarding signals into an external application, in our case, Unity3D. While BCI2000 is a fundamental tool for BCI research, its application layer is primarily designed for experimental use in laboratory settings. By having 3rd party graphical software communicate and supplement the native BCI2000 modules we gain the ability to create much more elaborate and complex BCI applications.

Unity3D

Unity3D is one of the most popular game engines, with over 700 million users [17], and is used primarily to create video games for numerous platforms including mobile devices, internet browsers, PCs and Macs. Other than game creation, recently Unity is being utilized to create videos, educational material, and even neuroscientific applications have been created with it [18]. Besides being useful in traditional screen-based video game design, native virtual and augmented reality support. Additionally, it uses C# as a dynamic scripting language for control of its various assets. In regard to routing signals from BCI2000 into Unity, the scripting component allows us to create network connections to read and write data streams. As mentioned in the previous section in regard to the telnet connection on the operator layer, when the Unity application is properly configured it will open BCI2000's operator module and will pass in a user-defined argument on what IP/port to connect to, as can be visualized in Figure 2. These connections remain open throughout the duration of the experiment and allow Unity to pass information into the operator module such as which signal source and processing modules to use, as well as various parameters. Two of BCI2000's parameters that are critical for using a 3rd party application layer are the *ConnectorInputAddress* and *ConnectorOutputAddress,* which takes as an argument an additional IP address and port number. By setting these arguments to a specific IP/port combinations, reading and writing state variable values to and from BCI2000 for use by the signal processing module can be accomplished.

For a simple motor imagery-based application we use the *ConnectorOutputAdress* to write the variables "TargetCode", "ResultCode", and "Feedback." By then setting the signal processing layer to look for these state conditions, we can use the *ConnectorInputAdress* to extract information pertaining to target hits, misses, aborts, and cursor position. This type of information is all that is needed from BCI2000 to create any

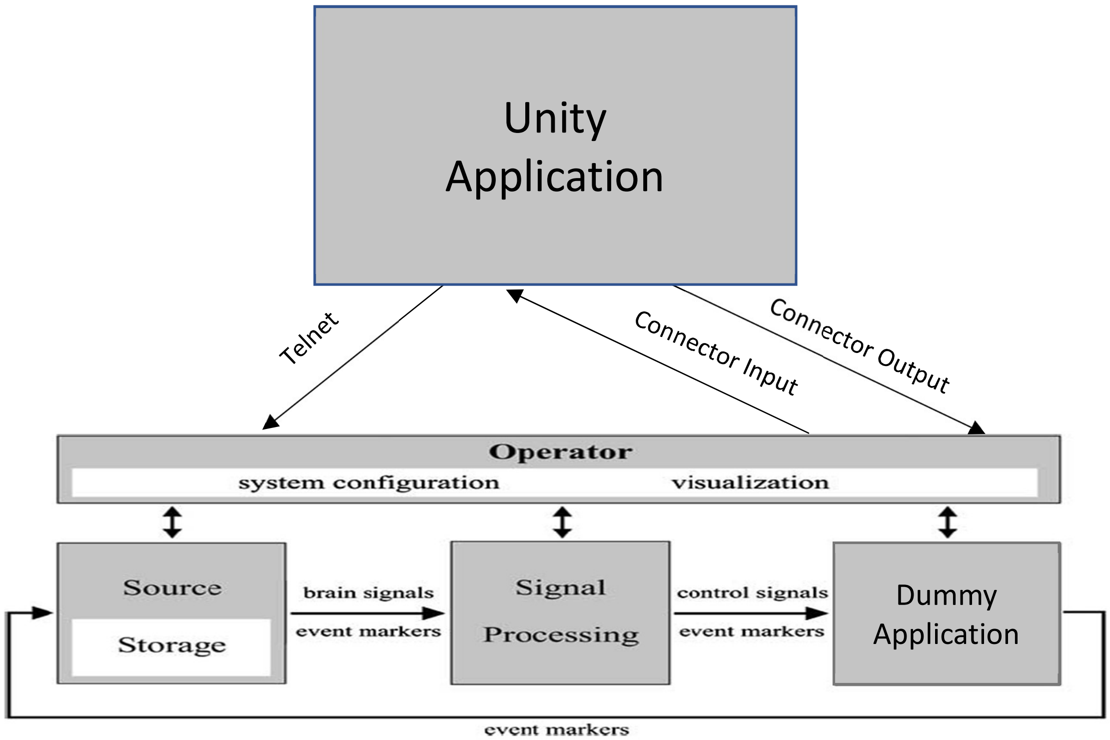

Figure 2: Unity/BCI2000 integration. Adapted from [1].

type of BCI-based application in Unity. By routing the signals from BCI2000 into Unity we can perform a one-to-one mapping between the control signal.

Unity utilizes what are called gameobjects, scenes, and components within its game engine. A gameobject can be any object (shape, model, user interface element) that holds values and components related to a particular scene. A scene is a collection of gameobjects that is presented to a user to interact with. Components can be added to gameobjects to provide functionality, such as C# scripts, position/rotation values, or pre-built functions, to name but a few. The creation of a Unity scene can therefore have gameobjects which have C# script components that provide a network connection to BCI2000, allowing for BCI control of virtual objects. A video of the creation of a Unity scene to be used with BCI can be found in the code's source repository given in the Discussion section. When C# scripts are attached to interactable elements, such as buttons, additional functionality is provided. For instance, when a button in the GUI (Figure 3) is referenced by a script, logic-based functions can be enacted, such as opening a new scene, or beginning a BCI-related task.

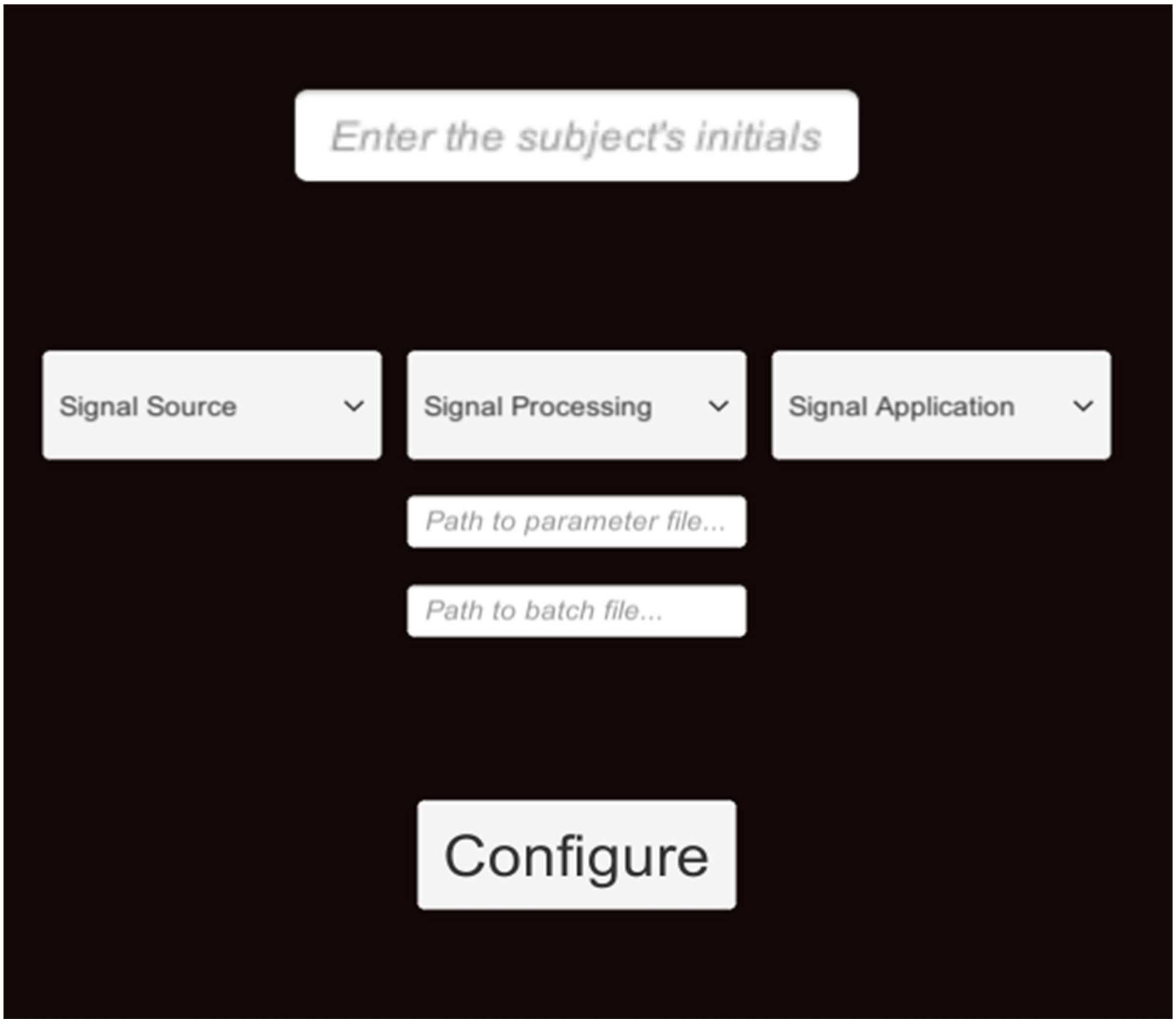

Figure 3: Graphical user interface for interacting with BCI2000 via Unity

When the corresponding dropdown menus and buttons are interacted with in the GUI, BCI2000 launches with its respective parameter files. After BCI2000 is configured, a new scene opens based upon the 'Signal Application" chosen from the GUI. If, for instance, a 1 or 2D motor imagery based scene was selected, game objects and scripts associated with that type of task will be presented. This includes configuration files to load the correct state variables for BCI2000 for this type of task, targets and a cursor to provide feedback to the user on the MI task, as well as a virtual camera for the user to view the scene and to move around within it. To customize the appearance of the scene the user can create their own gameobjects or models using software packages such as Blender or

Figure 4: Variants of an SMR-based Unity experiment scene

Maya. Additionally, numerous types of assets can be included in the scene such as trees, buildings, etc, to provide a more immersive experience, and many of these are free to use and available in the Unity Asset Store (several of which can be seen in Figure 4).

Implementation

Similar to BCI2000, the user can select which core modules and what parameter file to use, or they can select a preconfigured batch file. These commands are all written to BCI2000's operator layer as would be the case if using BCI2000's GUI. When the user presses the "Configure" button the commands are routed to BCI2000, BCI2000 loads the selected configuration, a new scene is loaded in Unity where the experiment takes place (Figure 4), and control is handed to the end-user. During this step that BCI2000 opens and is put in a resting state, ready to receive additional parameters. If the user selects a signal application template from the GUI (such as VR_LR_Candle) then the respective scene will load, and the experiment will begin whenever the user is ready.

Capability for the end-user to select when the application begins is enabled via keyboard or game-controller input. After the scene is loaded the user will be able to move around the environment. If the user is wearing a supported virtual reality device this will be by the corresponding controllers. If using the computer monitor for visual feedback, the user can move around via keyboard control or gamepad. This type of experimental setup is rare, if not unique, because the user is no longer presented with a static screen in with which to interact, but with a virtual environment that may contain elements not directly corresponding to the BCI task at hand.

Features

This software package was designed (but not limited) to be utilized in two ways. As all scenes, gameobjects, scripts, and other assets can be saved as assets of a project, assets can be created and shared amongst groups and individuals to be used as open-source templates. For instance, a basic SMR-based BCI task may include a cursor, two targets, and a script that communicates with the core BCI2000 software. If a user makes an addition to a particular template, or customizes the appearance of the targets/cursor, it can be repackaged and shared amongst others. Secondly, if a user just wishes to use the provided templates, little additional configuration is required.

To make this software applicable to as broad an audience as possible, support for VR and several IoT devices has been included as templates. These include the Leap Motion controller for hand tracking, Philips Hue API and TPLink to control supported lightbulbs/outlets, as well as the Roku API to control smart televisions. Currently the VR plugin used is SteamVR which allows this application to run using the HTC Vive, Oculus Rift, and Windows Mixed Reality devices. By not focusing on a single type of application or a particular physical device to be controlled, and offering basic support for multiple devices, it is hoped that more individuals build upon and share their contributions. This useful so others can easily reproduce experiments, as well expand upon them.

Templates

For the user's that simply want to use traditional BCI2000 tasks but with Unity's feedback stimuli, several templates have been included. These operate identically to traditional BCI2000 applications, the only difference being the user physically looks at Unity instead of BCI2000 (Figure 5). Currently these templates include an SMR-based left vs right paradigm, SMR-based up vs down paradigm, a SMR-based two-dimensional (up vs down vs left vs right) paradigm, and a P300-based speller. Furthermore, within the SMR-based paradigms, output can be routed to IoT devices and/or a VR headset.

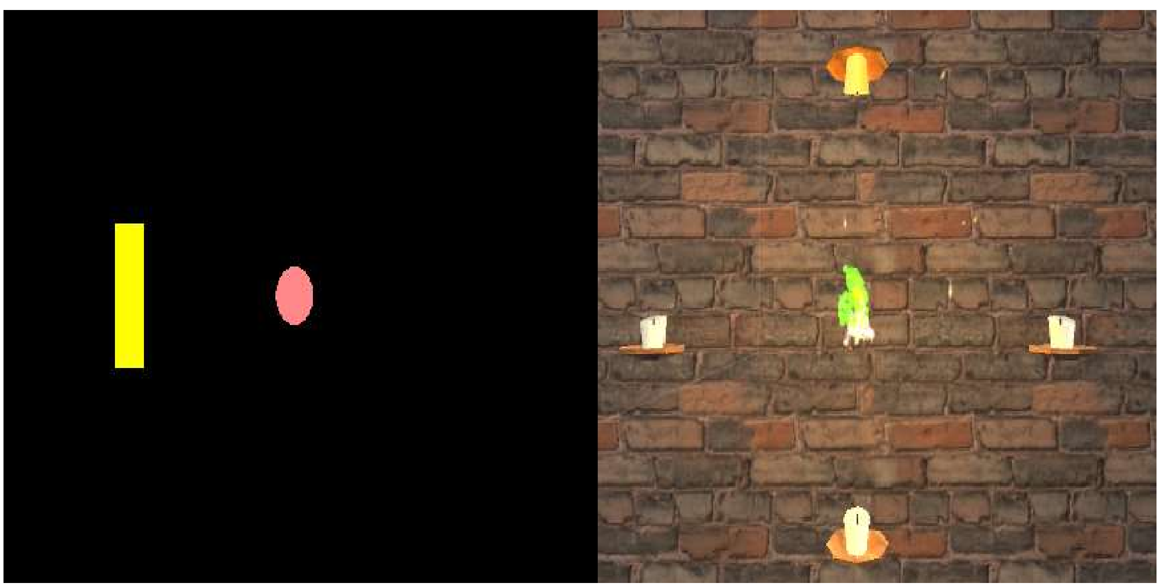

Figure 5: Side by side view of (L) BCI2000s native application layer and (R) Unity application layer

IoT

With the advent of the Internet of Things, almost any type of home device can be connected to the internet and controlled via a set of functions provided by the manufacturer. These devices range anywhere from toasters and lightbulbs to televisions and thermostats. By connecting to a local network the IoT device is assigned an IP address or given a token that allows users to interact with it via the manufacturer's official application (typically via smartphone), or a custom designed application. This is done by accessing the devices application programming interface (API). Users can interact with the API using hypertext transfer protocol (HTTP) requests to read and write data. For instance, if there are several IoT lightbulbs connected a home network a user can choose which one to connect to, turn it on and off, and set the brightness/hue values. For individuals with physical disabilities ranging from mild to severe, tasks which may appear simple to healthy individuals provide insurmountable obstacle towards independence and these types of services can be largely beneficial. What may seem like a small act to most people, turning a light on or adjusting the temperature, could make a world of difference for someone who only has the ability to type a command on their phone. For those that can't even do that, however, brain computer interfacing of things (BCIOT), no matter how simple, could make a huge impact, improving the quality of life and providing further independence from caregivers.

Three IoT devices have been integrated into this software package. Roku, for smart TV control, TP Link for lights/outlets, and the Philips Hue lightbulbs. As network security is a major concern for IoT devices, the IP/token parameters can, and must, be set by the individual administrators of their personal network. In order to control the IoT devices from within Unity the user opens the "IoT" scene provided as a template. After toggling a virtual switch to indicate which type of device is to be controlled, the user inputs the corresponding IP address/token, and the device automatically connects.

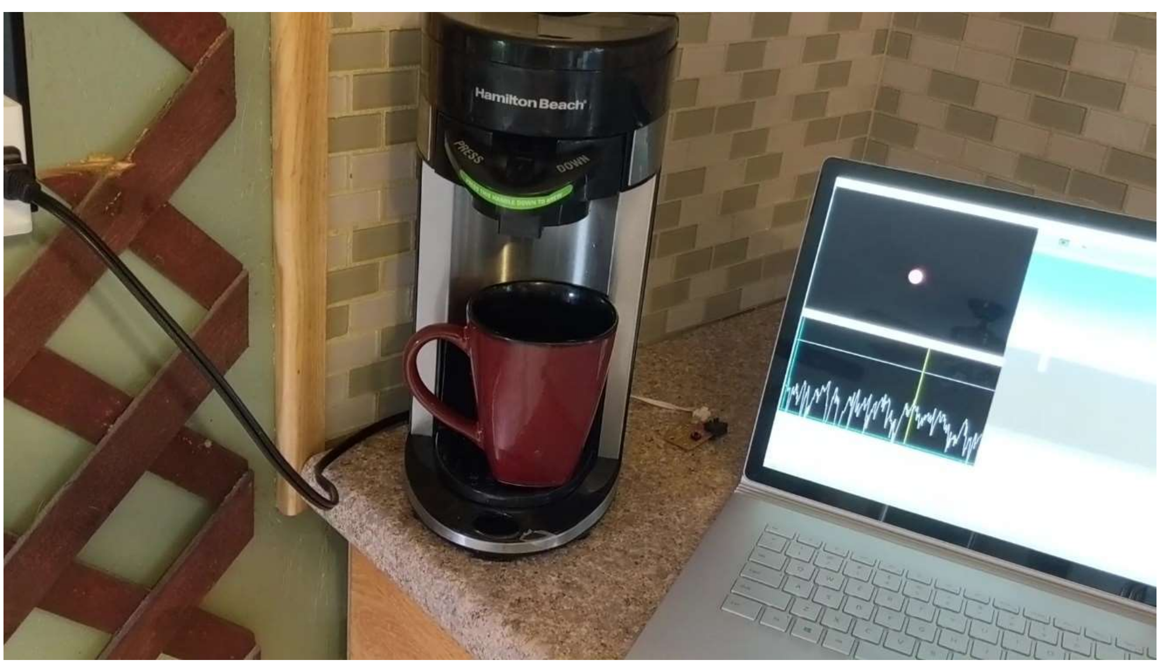

Figure 6: BCI controlled coffee pot via TP link interface

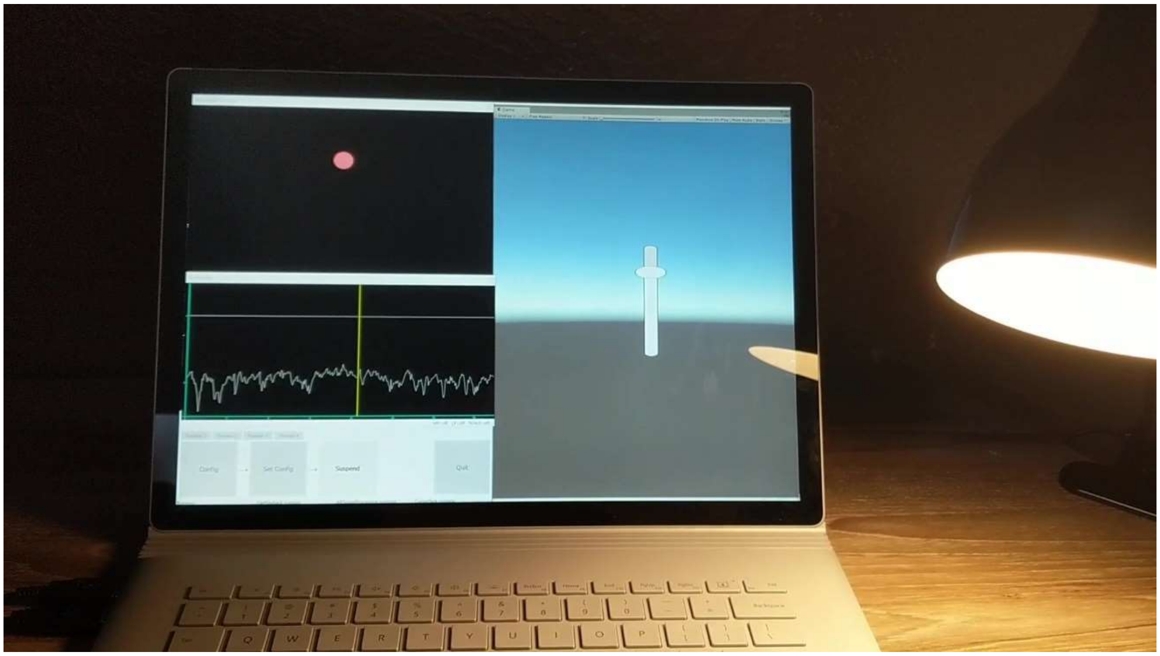

Figure 7: BCI controlled lightbulb via Philips Hue interface

All testing of BCI controlled IoT devices was completed offline using a subset of real-time SMR-based BCI data collected during the experiment protocol outlined later in this text. To simulate simple, asynchronous, control of IoT devices a one to one mapping was done between the control signal utilized in the up/down task of the VR study. To move the cursor up the user imagined opening/closing both hands simultaneously, and in order to move the cursor down the user was in a rested state, not imagining their hands moving at all. For control of a coffeepot via the TP Link smart switch (Figure 6) a threshold was experimentally determined and if the user performed bilateral motor imagery of the hands the coffeepot would turn on, and if they were in a rested state it would turn off. For control of a lightbulb via the Philips Hue bridge (Figure 7) there was a mapping between cursor position and the light's brightness intensity. The more the user thought "up" the brighter the light got. Lastly, for control of the volume using a TCL Roku-enabled smart TV there was a mapping between cursor control and volume (Figure 8). If the cursor position was greater than the value in the previous time point, the volume increased, if the cursor began to move down, the volume decreased. Other Unity gameobjects can be introducted into the scene such as buttons to provide on/off functionality to compliment the asynchronous control of the BCIOT device.

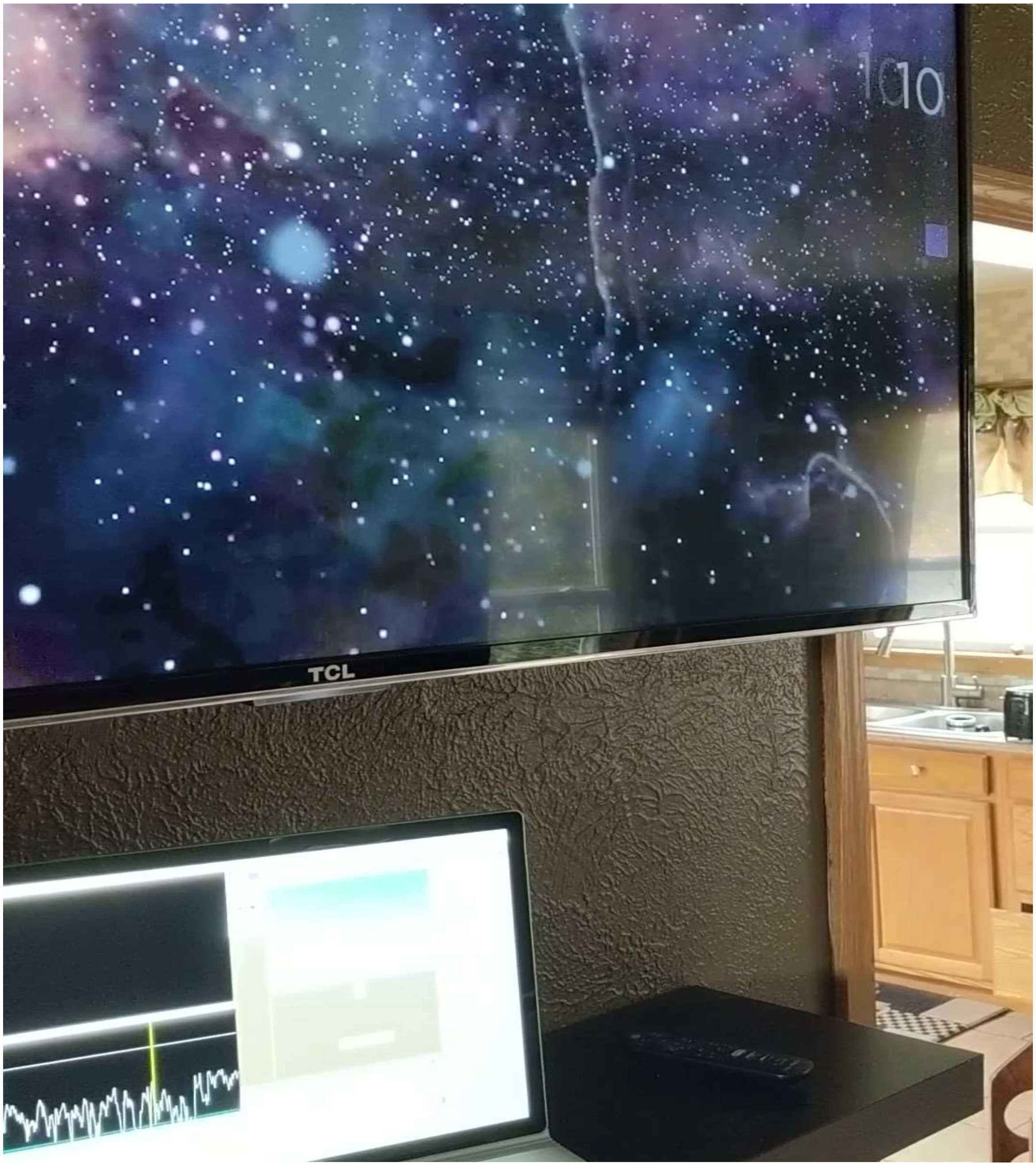

Figure 8: BCI controlled smart TV via Roku interface

Hybrid BCIs

By combining multiple BCI paradigms, and building a hybrid BCI, the information transfer rate (ITR) can be increased, which is critical if we are to build more complex BCI applications. The three BCI control signals of interest (SMR, SSVEP, P300) use independent electrodes [19], and can therefore be processed independently. A simple Laplacian-based filtering method for SMR-based BCI uses no more than 6 electrodes, whereas a reliable P300 or SSVEP signal can be acquired with an additional 7 or 8 electrodes each. By combining all three modalities (such as in Figure 10), additional control can be acquired by the end-user.

Figure 9: Example of hybrid BCI encompassing SSVEP, SMR, and P300-type applications

Additionally, by allowing multiple users to interact via BCI we can increase the complexity and create social experiments. This would also allow users to build confidence with relatively easy tasks and gradually increase the difficulty. This would be simple to accomplish by gamifying current experimental protocols such as the MindBalance [20] game. This software package allows multiple instances of BCI2000/Unity to be run together so collaborative or competitive applications could be created to test the effects of social interaction.

To accomplish multiple BCI2000 sessions, one of the flags the operator module can accept when initiating this telnet connection is “--AllowsMultipleInstances” which allows for various BCI2000 modules to be ran concurrently. With this flag checked it is possible to create an experiment receiving two separate neural data feeds from two individuals. Each instance of BCI2000 created with the corresponding BCI_Class C# script takes as an input an IP address and port number. Instances are stored in arrays so that each individual instance can be referenced and interact with all elements of a scene.

Mobile-use

In a traditional BCI experiment the computer serves two purposes: 1) processing and saving the user’s EEG data by physically interfacing with the neural acquisition hardware over a series of wires and 2) presenting visual stimuli via the computer monitor. With the advent of mobile EEG devices with onboard amplifiers that communicate to the computer via Bluetooth or WiFi we can remove the first case [21]. By presenting the visual stimuli on a similar wireless device, such as a smartphone, we can achieve a purely mobile-based BCI application. Figure 11 shows a mobile based BCI application with the optional use a VR headset to compliment it. It runs no differently than a computer-based BCI task except the visual stimuli is a Unity-based application built for a mobile device.

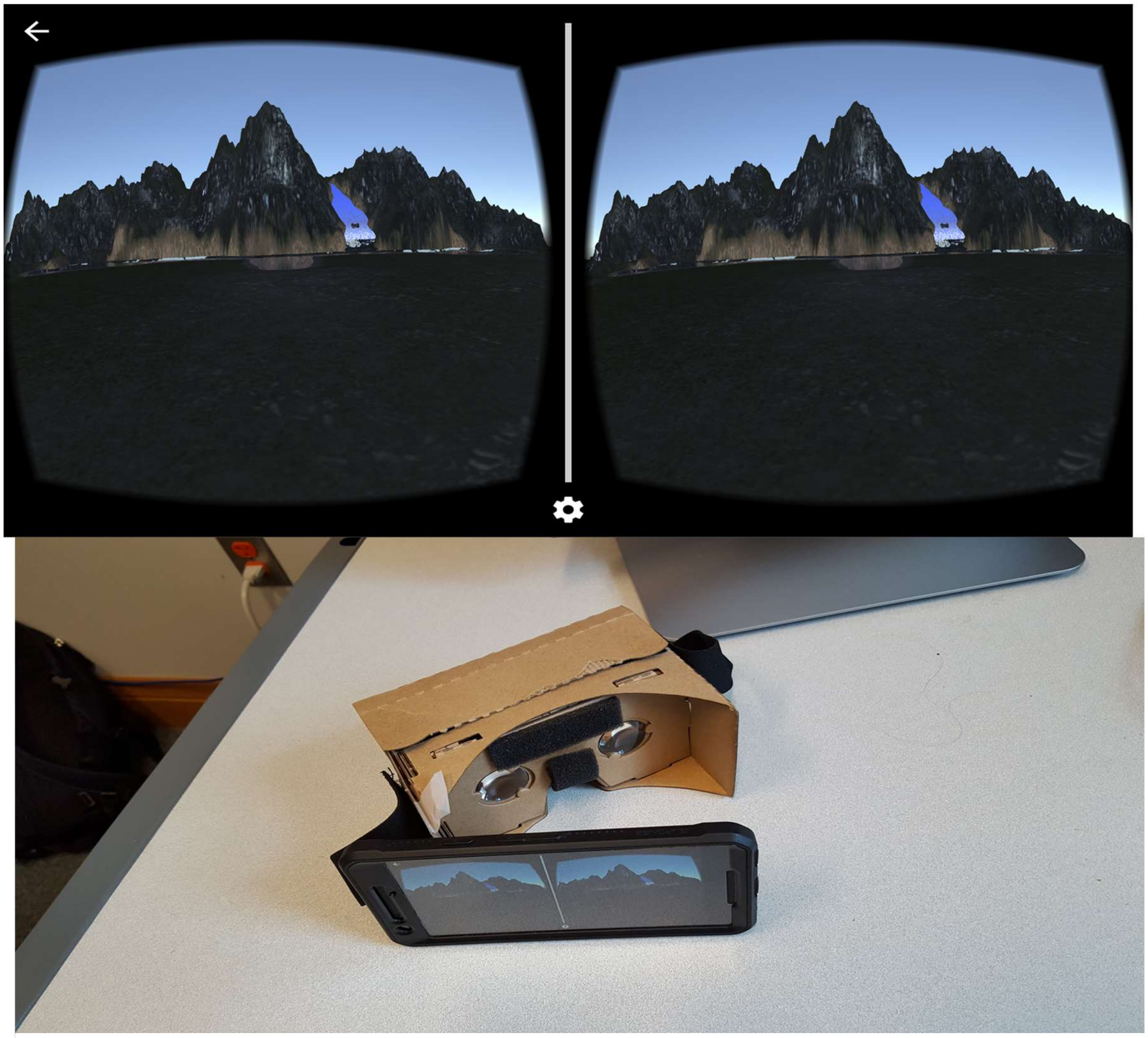

Figure 10: Mobile-based SMR-based BCI application. Bottom: Google Cardboard device for mobile-based VR applications

Virtual Reality

Unity's contributions as an application layer are evident with the inclusion for support of virtual/augmented reality libraries. Specifically, the SteamVR library for Unity allows for communication with multiple VR headsets. If the intended BCI application utilizes VR, the game camera will change various settings and allow the user to navigate the space (Figure 12) with a virtual reality controller. In an immersive VR application, users have shown to have an increased level of embodiment [22]. Coupling this type of interaction with a BCI could grant the user a higher level of control. In addition to the added level of embodiment, cognitive-control applications that are gamified [23] have the potential to improve attention and multitasking [24]. With the inclusion of the LeapMotion library for Unity user's can see a virtual representation of their physical hands. While this is useful for MI-based experiments, it can also be useful for psychological experiments akin to mirror treatments for phantom limb patients [25].

Figure 11: User controlling a SMR-based BCI via VR

Validation

To validate that the Unity-based BCI2000 application layer would work as intended, 31 subjects were recruited to test and validate the implementation of a virtual reality based, SMR-controlled, BCI experiment. The participant population was split into two group: those who experience only a single session, and those expose to multiple sessions. The first study compared the performance of users familiar with SMR-based BCI and users that were naïve in both a VR environment and a traditional SMR-based BCI task in a single session. The second study compared the learning rates of naïve subjects in a control group (traditional SMR-based BCI) and a VR group throughout 6 sessions.

The feedback hardware used in this study was the HTC Vive for the VR group and a 15-inch computer monitor for the control group. During the VR portion of the experiment users donned the VR headset, did a quick calibration of the lenses, and then spent 5 minutes walking around the virtual world via the tracked controllers to become accustomed to the experience and to determine whether they would have any adverse reactions to the visual stimulus.. When the users showed an adequate ability in traversing the 3D space and showed no signs of motion sickness, they were instructed to move into the experiment area. In order to give the subjects a level of autonomy not normally seen in BCI experiments, they were given control of how to position their virtual avatars which allowed them to view the motor imagery (MI) task from whatever perspective they chose (some chose to be less than 1 meter from the task while others chose to be several meters away). They were also given the ability to begin the experiment whenever they felt comfortable (instead of the experimenter dictating when the experiment would begin) by squeezing a button the tracked controller, and then placing the controller on the table in front of them. The control group simply indicated whenever they were ready to begin. The metric of performance utilized in this study was the percent valid correct (PVC) which is the number of hits divided by the total number of valid (non-timed out) attempts.

Data Acquisition

All participants were seated comfortably in front of a computer monitor, regardless of whether their group utilized it, for all sessions. All data was collected with a 64 channel EEG cap using a Neuroscan Synamps 2 amplifier, at a sampling rate of 1000Hz and filtered from .1-30Hz. A Laplacian spatial filter was used centered around electrodes C3 and C4. The Laplacian filter takes as input 10 electrodes, (C3/C4 and the 4 electrodes directly neighboring them) and sets a weight of 1 to C3/C3 and .25 to all others. By subtracting the neighboring electrodes from the electrodes of interest a better localized signal is attained. All signal acquisition and processing were identical in the two studies. For these motor imagery-based tasks, a target was presented on the screen/headset (as seen in Figure 11) and it was the participants goal to imagine opening and closing their respective hand(s) to move the cursor.

Study 1

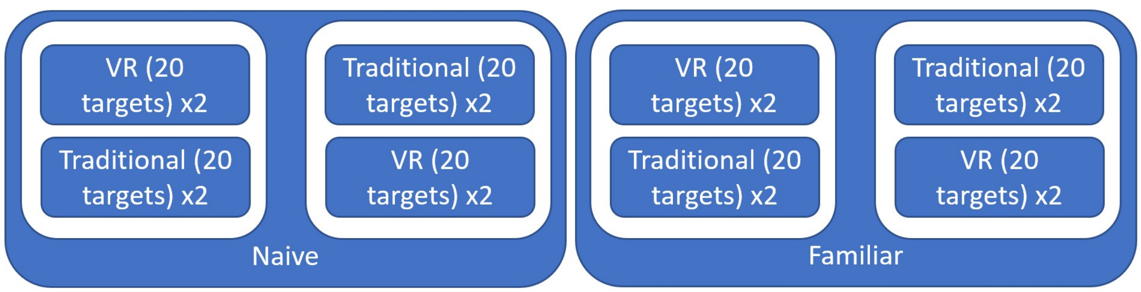

Figure 12: Study 1 pipeline

In the first study 22 subject were recruited, 15 having already participated in at least one BCI experiment, and 7 naive subjects. This study consisted of a single session primarily designed to determine whether or not there was a significant difference between the type of visual stimuli in non-naïve subjects. Each subject performed 2 two dimensional runs (20 targets/run that were randomly distributed between top, down, left, and right targets, with roughly 3 minutes per run) while receiving the visual stimulus in either VR or via the computer monitor (Figure 12). The subject had a 6 second period where the target was presented. If no target was hit during this period, the trial was aborted, if a target was hit, the trial would end early. Regardless of a hit, miss, or aborted trial, the participant was then given a 2 second inter-trial rest period before the next target was presented.

Figure 13: Study 1 visual feedback

After the 2 initial runs, subjects completed an additional 2 runs in the alternate paradigm (switch between VR and computer monitor). After a 5-minute waiting period the subjects repeated both tasks in the previous order (if they started with a VR stimulus in the first half, they again started with a VR stimulus in the second half). The starting stimuli (VR or traditional) was randomized so that each group had an equal distribution of individuals who began with both the VR and computer monitor stimuli. In each group the filtering pipeline was identical, with the only difference being the visual stimuli that was presented

Study 2

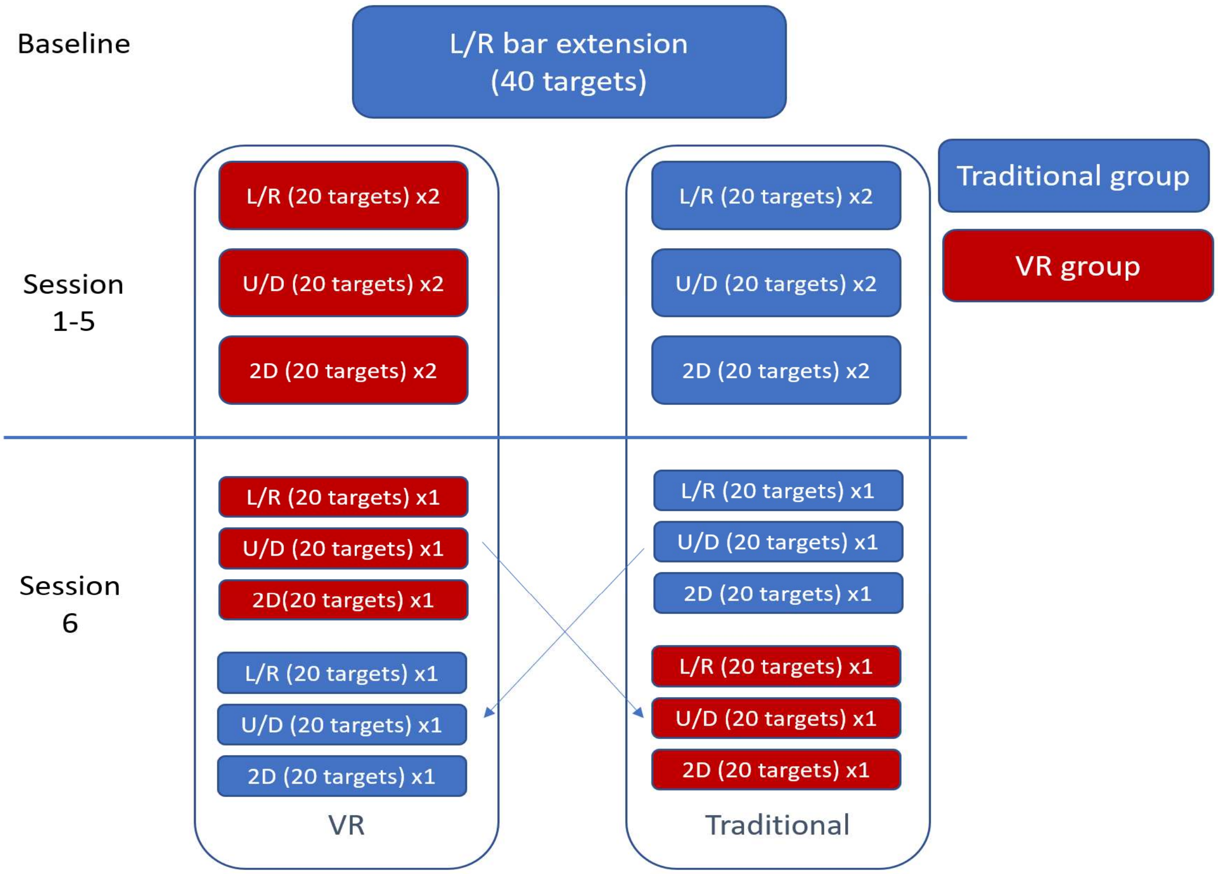

Figure 14: Study 2 pipeline

In this study 9 naive subjects were recruited and participated in 6 sessions each. This study was designed to test the performance/learning rates between naïve subjects receiving a visual stimulus of targets presented via the computer monitor or VR headset. The first task presented to the users was a motor-imagery feedback task. Users were presented with a black screen on the computer monitor and asked either to physically open/close their respective hands or to imagine the act. This was to measure their baseline EEG activity prior to the BCI task.

Figure 15: Study 2 visual feedback

In the second study, each session consisted of 3 left vs right runs (20 targets per run, roughly 3 minutes per run) followed by 3 up versus down runs, and finally 3 two dimensional runs. After this initial phase the subjects were given a 5-minute rest period before repeating the task a second time. This pipeline was used for the first 5 of 6 sessions Figure 14). When the participants returned for the 6th and final session they were once again presented with an identical protocol to the first 5 sessions, but after the 5-minute rest period and completion of the first half of the session, they were switched to an alternate paradigm. If during the first 5 sessions the participants were presented with targets via a computer monitor, for the second half of the 6th session they would be presented with targets via a VR headset (Figure 15). The opposite was true of the other group (if original stimuli was VR, novel stimulus was provided via the computer monitor). This test was designed to assume that after 5 sessions of a MI-based BCI task, all participants were relatively proficient in the task. With a novel stimulus introduced it would show whether or not either stimulus had any affect over the other.

Results

As can be seen in the figures below, performing MI-based BCI tasks in VR does not affect one’s ability to perform the task. Across both studies and 31 subjects assessed, mean differences were not scene between groups, indicating 1) signal transmission from BCI2000 to Unity and back again did not introduce any noticeable latency, and 2) that users performed no worse when in an immersive, virtual reality, BCI experiment.

While not statistically significant, one interesting trend that may offer additional information is in Figure 17. There appears to be an increase in performance while switching from the control stimulus to the VR stimulus in the group that has already had some introduction to the SMR-based BCI task in the past. This can be compared with Figure 19 in study 2. Once again, switching from the control stimulus (of which this group was trained in for 5.5 sessions) to the VR stimulus shows a similar increase in performance, not seen in both study 1 and study 2’s participants going from a VR to a control stimulus. While not significant, it appears to be consistent throughout both studies.

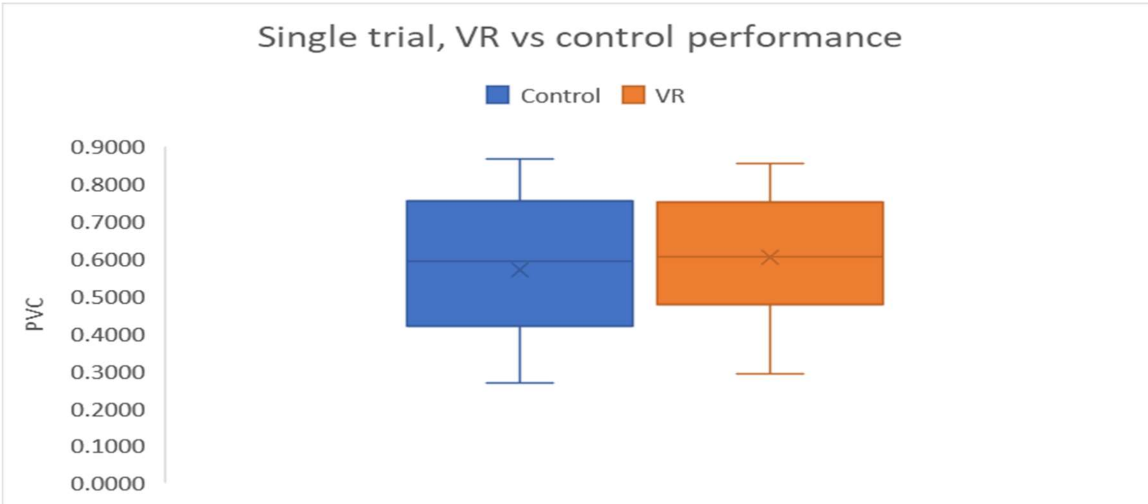

Figure 16: Results from study 1 indicating no change in performance between groups

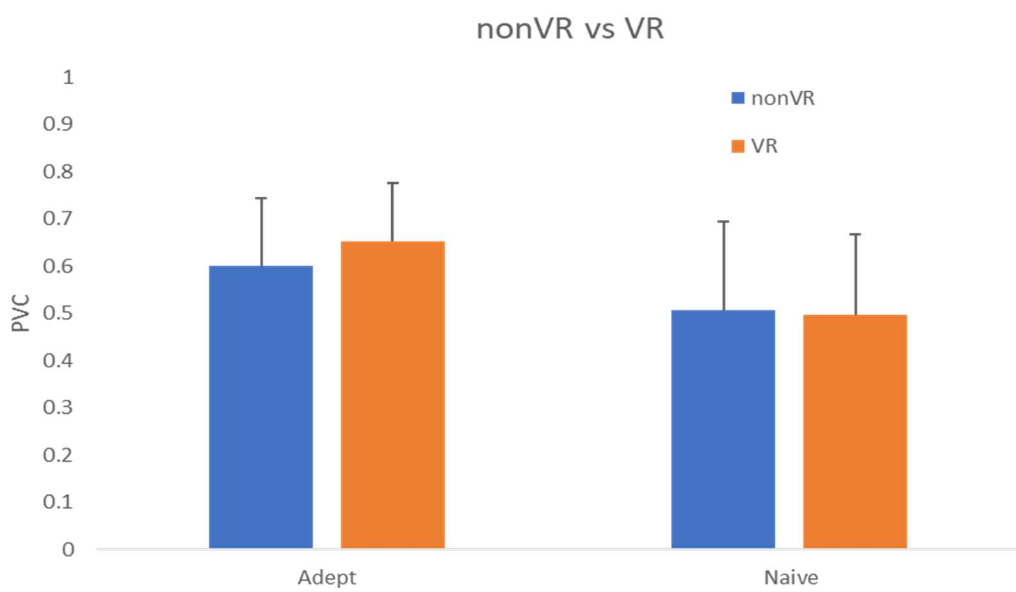

Figure 17: Results from study 1 showing a small increase in performance in adept subjects switching from a control stimulus to a VR one

Additionally, there was no significant difference in between groups switching from a VR stimulus to a control stimulus, further indicating the visual stimuli alone does not impede performance.

Figure 18: Results from study 2 indicating no significant difference in performance between groups

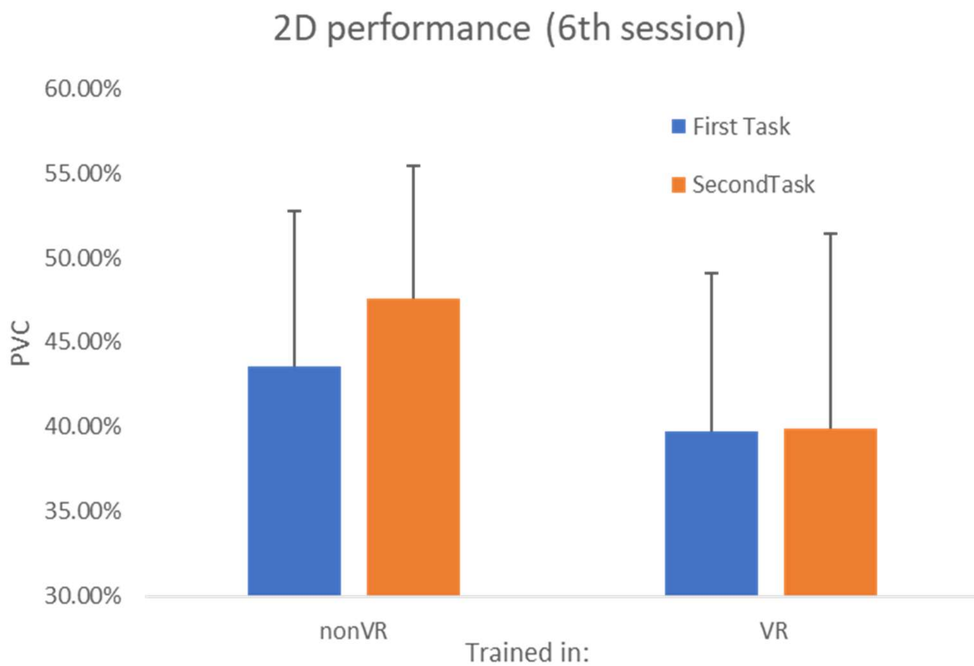

Figure 19: Results from study 2 showing a small increase in performance in adept subjects switching from a control stimulus to a VR one

Discussion

With the pace of technological advancement exponentially increasing, amazing strides in virtual/augmented reality and the internet of things have been seen in the past half-decade. Just as advances in these fields required prerequisite technological improvements in network infrastructure and high-end consumer graphics, so to does the continued improvement of BCI use rely on a prerequisite set of technologies. While research-grade neural acquisition devices have enabled a vast trove of reliable datasets to be analyzed, consumer-grade EEG/BCI devices are just now beginning to come to market. Machine learning and artificial intelligence, for instance, was founded within research labs and evolved outside of it, the same may be true of the BCI field.

While this application layer has been built for use with BCI2000, there are no technical limitations as to why it could not be used in conjunction with other BCI systems

(OpenVibe, OpenBCI, etc) since the reading and writing of data is done using simple TCP/UDP communication. Likewise, example scenes include support for the Philips Hue and Roku API, but there are no technical limitations to expand beyond this as they only require HTTP requests.

Upon publication the accompanying source code distribution will be found at, *github.com/bfinl/UnityBCI*, under a General Public License (GPL). Users are encouraged to download, distribute, and make contributions. While the current state of this release does require a fair amount of manual tuning to begin creating virtual scenes to integrate with BCI2000, it does support all acquisition and processing modules and includes a set of core BCI2000 modules. Due to the open source nature of this application a user can fork the code, create an application, push the updated code, and grant other users access to it. Because Unity can run in Editor-mode, by importing the newly created assets from a different user, no compilation is necessary, so experiment/scenes can be shared and used instantly. It is hoped that by offering an easy-to-use tool to create BCI applications, the field will grow faster.

While these newer technologies could provide freedom to explore advances in BCI research, several issues may prevent it from becoming wholly adopted. For instance, VR use greatly expands on the level of embodiment a user feels during their session, however it yet to be seen whether this embodiment will be more motivation, or distracting. Similarly, while the majority of subjects reported no sense of motion sickness or vision impairments during prolong VR use, one subject was excluded from the study due to discomfort in the first session. Because both a VR headset and EEG electrodes are placed directly on top of the head, these two can physically interfere with each other. As can be seen in Figure 11, the HTC Vive's straps directly press on electrodes C3/C4, the primary electrodes used in SMR-based BCI control. Lastly, the VR device adds an additional weight upon the user's head and may cause fatigue not seen in other groups. All user's in these studies were given a 5-minute break halfway through the session, regardless of group, to minimize this external factor.

Conclusion

Using widely available software packages and network calls it is possible to route BCI control signals into various applications such as game engines, virtual reality devices, and personal home devices. These devices can be controlled, and environments traversed, using a combination of BCI control and physical input. This greatly expands the potential for creating complex BCI applications that have not been reported in the literature. While user performance did not increase during the immersive VR task, users within the group showed a comparable learning rate relative to their counterparts in the control group, indicating that there are no detrimental effects due to latency of network transmission or cognitive aspects such as motivation or distraction.

The work presented here describes the creation, implementation, and validation of a Unity game engine-based application layer for BCI2000. Supplanting the native application layer allows for the integration of advanced 3D graphics, VR, and IoT devices. This software package includes several template scenes for rapidly implementing various BCI paradigms along with their use in VR or IoT applications. If user's wish to build there own BCI applications, or build upon published templates, an easy interface allows them the freedom to edit each individual component from the visual stimuli to the control signals of interest.

Bibliography

[1] Millán, J. D. R., et al. "Combining brain–computer interfaces and assistive technologies: state-of-the-art and challenges." *Frontiers in neuroscience* 4 (2010).

[2] Leeb, Robert, et al. "Combining BCI and Virtual Reality: Scouting Virtual Worlds." *Toward brain-computer interfacing*(2007): 393.

[3] Faller, Josef, et al. "An application framework for controlling an avatar in a desktop-based virtual environment via a software SSVEP brain–computer interface." *Presence: teleoperators and virtual environments* 19.1 (2010): 25-34.

[4] He, Bin, editor. *Neural Engineering*. 2005 edition, Springer, 2005.

[5] Machado, Sergio, et al. "EEG-based brain-computer interfaces: an overview of basic concepts and clinical applications in neurorehabilitation." *Reviews in the neurosciences* 21.6 (2010): 451-468.

[6] Leeb, Robert, et al. "Self-paced (asynchronous) BCI control of a wheelchair in virtual environments: a case study with a tetraplegic." *Computational intelligence and neuroscience*2007 (2007).

[7] Yuan, H., et al. "Cortical Imaging of Event-Related (de)Synchronization During Online Control of Brain-Computer Interface Using Minimum-Norm Estimates in Frequency Domain." *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 16, no. 5, Oct. 2008, pp. 425–31. *IEEE Xplore*, doi:10.1109/TNSRE.2008.2003384.

[8]Yuan, H., and B. He. "Brain #x2013;Computer Interfaces Using Sensorimotor Rhythms: Current State and Future Perspectives." *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 5, May 2014, pp. 1425–35. *IEEE Xplore*, doi:10.1109/TBME.2014.2312397.

[9] Royer, A. S., et al. "EEG Control of a Virtual Helicopter in 3-Dimensional Space Using Intelligent Control Strategies." *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 18, no. 6, Dec. 2010, pp. 581–89. *IEEE Xplore*, doi:10.1109/TNSRE.2010.2077654.

[10] Meng, Jianjun, et al. "Noninvasive Electroencephalogram Based Control of a Robotic Arm for Reach and Grasp Tasks." *Scientific Reports*, vol. 6, Dec. 2016. *PubMed Central*, doi:10.1038/srep38565.

[11] Edelman, B. J., et al. "EEG Source Imaging Enhances the Decoding of Complex Right-Hand Motor Imagery Tasks." *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 1, Jan. 2016, pp. 4–14. *IEEE Xplore*, doi:10.1109/TBME.2015.2467312.

[12] He, B., et al. “Noninvasive Brain-Computer Interfaces Based on Sensorimotor Rhythms.” *Proceedings of the IEEE*, vol. 103, no. 6, June 2015, pp. 907–25. *IEEE Xplore*, doi:10.1109/JPROC.2015.2407272.

[13] Lécuyer, Anatole, et al. "Brain-computer interfaces, virtual reality, and videogames." *Computer* 41.10 (2008).

[14] Ron-Angevin, Ricardo, and Antonio Díaz-Estrella. "Brain–computer interface: Changes in performance using virtual reality techniques." *Neuroscience letters* 449.2 (2009): 123-127.

[15] Wu, Hong, et al. "Evaluation of Motor Training Performance in 3D Virtual Environment via Combining Brain-computer Interface and Haptic Feedback." *Procedia Computer Science*107 (2017): 256-261.

[16] Schalk, Gerwin, et al. "BCI2000: a general-purpose brain-computer interface (BCI) system." *IEEE Transactions on biomedical engineering* 51.6 (2004): 1034-1043.

[17] https://unity3d.com/public-relations

[18] Mullen, T., Kothe, C. Chi, Y.M., Ojeda, A., Kerth, T., Makeig, S., Cauwenberghs, G., Jung, T-P. (2013) Real-Time Modeling and 3D Visualization of Source Dynamics and Connectivity Using Wearable EEG. 35th Annual International Conference of the IEEE Engineering in Biology and Medicine Society.

[19] Edlinger, Günter, Gunther Krausz, and Christoph Guger. "A dry electrode concept for SMR, P300 and SSVEP based BCIs." *Complex Medical Engineering 2012*

[20] Lalor, Edmund C., et al. "Steady-state VEP-based brain-computer interface control in an immersive 3D gaming environment." *EURASIP journal on applied signal processing* 2005 (2005): 3156-3164.

[21] Navarro, Karla Felix. "Wearable, wireless brain computer interfaces in augmented reality environments." *Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on*. Vol. 2. IEEE, 2004.

[22] Kilteni, Konstantina, Raphaela Groten, and Mel Slater. "The sense of embodiment in virtual reality." *Presence: Teleoperators and Virtual Environments* 21.4 (2012): 373-387.

[23] Bayliss, Jessica D., and Dana H. Ballard. "A virtual reality testbed for brain-computer interface research." *IEEE Transactions on Rehabilitation Engineering* 8.2 (2000): 188-190.F

[24] Anguera, Joaquin A., et al. "Video game training enhances cognitive control in older adults." *Nature*501.7465 (2013): 97-101.

[25] Chan, Brenda L., et al. "Mirror therapy for phantom limb pain." *New England Journal of Medicine* 357.21 (2007): 2206-2207.