

Dartmouth College

Dartmouth Digital Commons

Dartmouth College Undergraduate Theses

Theses and Dissertations

6-1-2018

Navigating Virtual Reality Using Only Your Gazes and Mind

Christopher J. Kymn
Dartmouth College

Follow this and additional works at: https://digitalcommons.dartmouth.edu/senior_theses



Part of the [Computer Sciences Commons](#)

Recommended Citation

Kymn, Christopher J., "Navigating Virtual Reality Using Only Your Gazes and Mind" (2018). *Dartmouth College Undergraduate Theses*. 138.

https://digitalcommons.dartmouth.edu/senior_theses/138

This Thesis (Undergraduate) is brought to you for free and open access by the Theses and Dissertations at Dartmouth Digital Commons. It has been accepted for inclusion in Dartmouth College Undergraduate Theses by an authorized administrator of Dartmouth Digital Commons. For more information, please contact dartmouthdigitalcommons@groups.dartmouth.edu.

Dartmouth Computer Science Technical Report TR2018-857: Navigating Virtual Reality Using Only Your Gazes and Mind

Christopher Kymn
Dartmouth College, Department of Computer Science

June 1, 2018

Abstract

We present a novel brain-computer interface that allows users to control virtual reality using only their brain waves and eye gazes. The interface allows users to control multiple objects with two dimensions of control. The system is portable, non-invasive, and runs on commercial-grade hardware. It thus provides a high-transmission and user-adaptive interface for users to engage in virtual reality. In addition, we present a training procedure that allows the user to increase control over the brain-computer interface by engaging with the program in an intuitive manner. We explain this procedure and demonstrate its effectiveness in formulating more readily interpreted commands.

1 Introduction

A brain-computer interface (BCI) is a device that directly reads the activity of the brain in order to control an electronic device. For example, while a keyboard requires manual input, the BCI equivalent would attempt to decode keystrokes from patterns of neuronal activity. The possible utility of BCIs is constantly growing. Such devices have medical applications, enabling communication for those who have no other ability to communicate with the outside world [3], but they can be useful in other ways: such as when one wishes to communicate silently with the outside world or the BCI is less cumbersome than the equivalent manual interface. [7], for exam-

ple, studied the prospects of a BCI that decoded brain activity into speech sequences.

While BCIs offer a powerful framework for augmenting user control, they suffer from several limitations that hamper their functionality. First, a large number of BCIs require either invasive procedures or professional medical setup in order for the device to properly work. This process limits the accessibility of the device as a widespread tool. Second, many BCI implementations rely on passive, rather than active processes. For example, some BCIs, which focus on decoding the visual stimulus a user attends to, thus end up limiting user options to the type of stimuli that are presented. Finally, BCI controls have often only allowed control of one single object, also limiting the range of options that a user may have available to her.

A main contribution of this paper is to present a BCI system that jointly addresses these issues. The system utilizes the Emotiv EPOC (Figure 1), a commercial-grade electroencephalogram (EEG) that is non-invasive and does not require medical attention to set up. The program affords the user active control over directional movement, which allows for active, rather than passive gameplay. Finally, the system is integrated with the Fove headset which allows for gaze-tracking, creating an immersive gaming experience in virtual reality. To the best of our knowledge, this is the first demonstration of multiple-object control in a brain-computer interface with gaze-tracking

The other main contribution of this paper is to present a training algorithm such that the

user can learn to control the brain-computer interface. The convenience and portability of the Emotiv EPOC come along with the drawbacks of being less powerful than medical-grade commitment. Therefore, the success of the proposed BCI is augmented by a training procedure that boosts a user’s ability to control her brain waves. While previous studies have developed similar training procedures, the application to this headset and implementation for this headset are new.



Figure 1: The Emotiv EPOC Device.

2 Background and Theory

Relevance of EEG to Virtual Reality. The primary BCI used for achieving control in virtual reality is an electroencephalogram (EEG), a standard device in neuroimaging. An electroencephalogram detects the electrical activity off of the surface of the brain at designated sensors, which allows the procedure to be non-invasive. The EEG has the advantage of high temporal resolution but has relatively poor spatio-temporal resolution, since it is constrained by the number of sensors placed on the scalp and only sums up large sums of electrical activity [2]. The Emotiv EPOC is an EEG with 14 sensors and 2 reference sensors (Figure 2) compared to the 64 sensors that a standard medical-grade EEG would have.

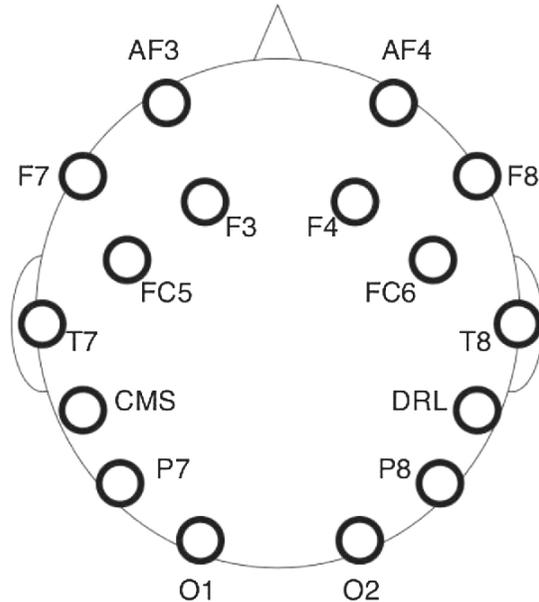


Figure 2: locations of EEG sensors for Emotiv EPOC.

Despite some of the difficulties involved in the EPOC, such as its lower spatial/temporal resolution compared to clinical devices, past demonstrations have shown promise with the device. [19] showed that the EPOC can be used to detect motor imagery, and Emotiv introduced a Cognitiv suite that uses training data. Unfortunately, the algorithms behind this suite are proprietary, and one study found that users were only able to achieve 36% accuracy (for 1-dimensional control, with a neutral option) after training [9]. My algorithm, by contrast, attempts to achieve a higher degree of accuracy for 2-dimensional control.

The EPOC is a promising candidate for implementing a BCI that works with virtual reality. Some benefits that a BCI can offer to virtual reality include a higher capacity for rapid transmission of information, ability to monitor user engagement via monitoring brain waves, and enabling control for users who cannot interact with ordinary gaming controls. By virtue of being a portable, non-invasive EEG that has relatively simple setup instructions, the EPOC could make these benefits more easily realizable for more users.

Methods of EEG Systems. The type of

EEG system exploits differences in the sensorimotor rhythms in different movements in order to classify between different signals. The current neuroscientific explanation for the signal that these interfaces detect is that when movement is imagined or actualized, the motor cortex signals synchronize, usually meaning resulting in a decrease in power per a Fast Fourier Transform. This change in pattern can then be detected and measured.

Previous studies, such as [12], have shown that imagined movements, rather than actual movements, are sufficient to generate activity. This means that the user need not actually move a muscle in order for the signal to change. [22] found that some users initially initiated movement when initially using the BCI, but were able to gain control of the device without moving muscles during later runs, as demonstrated by the lack of electromyographic (EMG) activity which occurs during muscle activity.

These differ from algorithms which attempt to determine what the user is looking at. Steady State Visual Evoked Potential (SSVEP) based BCIs determine whether a user is looking at a stimulus based on the frequency of firing patterns in the BCI [20]. P300 algorithms detect a spike in activity for around 300ms after a user views a significant stimulus [5]. However, since gaze tracking already yields information about what the user is looking at, we opt to instead track motor commands which can augment information based off of gaze tracking.

3 Problems with Robustness

One difficulty in working with EEG is the low signal-to-noise ratio. On one hand, the spatial resolution of a typical EEG is low: compared to a fMRI scan which can record brain activity at a resolution of 2mm, medical applications of EEG only have 64 sensors, placed on the scalp, which makes it difficult to localize different sources of activity. Second, due to the time-varying signal of the EEG, the dimensionality of the EEG data is relatively high. (If there are 14 channels processed for 1 second at 128 Hz, there are 1728

dimensions.) Although recent methods in "deep learning" [10] have made it easier to extract high-level patterns from the data, they require a large training set to achieve good performance, which might be a daunting ask for a dedicated user.

While good algorithm design and feature selection, described in the next section, may make it easier to extract the helpful signal from a BCI, a second difficulty arises from non-cognitive artifacts that disrupt the EEG signal. The most conspicuous disruption is blinking. Personal experiments found that it disrupts 10 of 14 EPOC channels by registering a short spike hundreds of μV higher than the typical range of activity (-100 to 100 μV). While this disturbs the integrity of the signal, blinks are relatively easy to detect and classify. Although methods have been developed in order to eliminate blinking altogether from datasets, we recognize that blinking is a natural and likely occurrence in actual gameplay; thus training data should be able to operate without blinking being taken into account. The algorithm section below utilizes gaze tracking to detect blinks and prevent them from interfering in gaze control.

4 Algorithm Design

4.1 Overview

Figure 4 summarizes the control flow of the program at any given time-step. At each interval (currently set to 336 ms), the program takes as input data from both the Fove and Emotiv headsets. The program uses the EEG data in order to determine which command (or lack thereof) to send to an object in the virtual reality system. Meanwhile, the Fove data is used for two purposes: to determine if there was a blink detected by the system, and to determine which object the user is looking at, if any. If 1) it is detected that a user is looking at an object, 2) the user did not blink during the time period of the study, and 3) the user sent a command to an object, the program sends the appropriate command to the virtual reality system. A command is operationally defined as a direction and a magnitude for the user to move. Due to the current

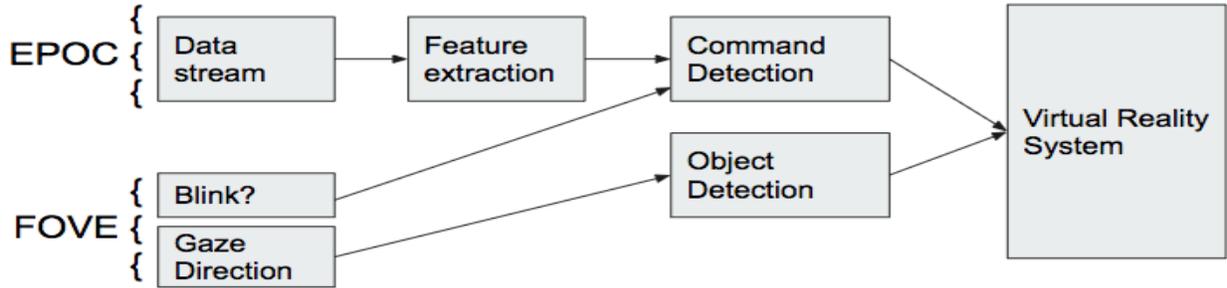


Figure 3: Control flow of the program.

cycles of the program, the user can send up to 3 commands per second.

4.2 Motion Command Detection

Preprocessing At each sample, we subtract the raw data (in μV) by 4200 to offset the values in the EEG. In order to refine the data and potentially remove the effect of other EEG activity, we then use common average referencing (CAR) as a preprocessing step. CAR subtracts the average value of sensors at a given point in a time series, thus effectively yielding the voltages relative to other sensors. Since we only focus on one region, the

Feature Extraction We consider the activity over 4 sensors: F3, F4, FC5, and FC6. The activity of these sensors in particular is helpful as they are the ones placed closest to the motor cortex, in order to determine the accessibility and ease of use in the headset. It is worth noting that these sensors are not at the optimal location, since studies such as [22] with 64 EEG instead evaluated C3/C4, which are directly over the motor cortex. [4] went as far as to rewire the headset and build a new shell in order to place these electrodes in a better location. While this is certainly a valid approach, we attempted to maintain the Emotiv EPOC as-is and instead learn a statistical relation between other features and the sensorimotor rhythm.

In order to get the activity of the features, we convert the information from the time domain to the frequency domain using a Fast Fourier transform. The Fast Fourier Transform decomposes the time series from the previous 43 sam-

ples into a set of frequency bands. Since it evaluates the last third of a second, the resolution of the Fourier transform is approximately 3 Hz. We use the mu and beta rhythms (approximately 12 and 24 Hz, respectively) as features in the classification algorithm. These are the common ones in the literature and relate to effects observed in sensorimotor rhythms [22]. Studies have shown that users are able to adaptively control these parameters over time.

Classification Algorithm At this stage, we have eight total features: the beta and mu rhythms for four channels: F3, F4, FC5, and FC6.

We use two simple linear regression algorithms in order to determine the direction to move. Each algorithm controls one dimension of movement. That is, the first algorithm outputs a classification value of up or down, while the second algorithm outputs a classification value of left or right. Therefore, it is possible for the user to move in two directions in one step.

Similar to [22], we use the difference in mu rhythms between the two channels on the left hemisphere of the brain (F3, FC5) and the right hemisphere of the brain (F4, FC6) to determine horizontal movement. This wiring allows the user to intuitively gain control of the brain computer interface by thinking different commands. In order to move left, the user can think of moving the left side of their body. Imagination of activity on the left side of the body leads to a temporary decrease in the mu rhythm in the contralateral side, that is, the right hemisphere. Since this decrease is more strongly detected by F4 and FC6, the difference in commands is de-

tected by the algorithm. A similar explanation can be given for controlling the right hand side, with an increase in right hand activity resulting in a decrease in left hemisphere activity.

We use the sum of the beta rhythm power of the four channels in order to determine vertical movement. This wiring also yields a potentially intuitive interpretation: higher levels of imagined activity are connected to upward movement, whereas lower levels of activity are connected to downward movement. Note the use of mu, rather than beta, power in order to establish independence of vertical from horizontal movement.

Therefore, the horizontal and vertical classification algorithms can be described by equations 1 and 2, respectively:

$$M_h = w_{F3,\mu}P_{F3,\mu} + w_{FC5,\mu}P_{FC5,\mu} + w_{F4,\mu}P_{F4,\mu} + w_{FC6,\mu}P_{FC6,\mu} + b_h \quad (1)$$

$$M_v = w_{F3,\beta}P_{F3,\beta} + w_{FC5,\beta}P_{FC5,\beta} + w_{F4,\beta}P_{F4,\beta} + w_{FC6,\beta}P_{FC6,\beta} + b_v \quad (2)$$

All weights are initialized to 1 except $w_{F4,\mu}$ and $w_{FC6,\mu}$, which are initialized to -1. b_h and b_v are bias terms, initialized according to the first step of the training procedure, described below. For positive values of M_h and M_v , the system moves left and down respectively; otherwise, the system registers the command as right and up.

Training Procedure In order to adapt the weights of each channel to the individual user and set biases, we develop a set of training programs so that the user can learn to control the user’s activity.

First, the user attempts to train each individual command: left, right, up, and down. The program prompts the user to think one of the four commands and logs the EEG data for 3 seconds. This information is used to set the bias terms. b_h and b_v are set to the average value of the weighted sums of the horizontal and vertical power features, respectively.

Second, the user plays a training game, which is simplified to allow the user to practice control over parameters. The game is simplified in two

senses: the user only controls one object (no gaze tracking is involved yet), and the user focuses on controlling the object in only one dimension. During each trial, the goal of the user is to move the object from the center of the screen to an edge of the screen, set at random. Since the optimal strategy is to only make moves in the appropriate direction, the recorded gameplay can be taken as ground-truth attempts to move in the selected direction. The gameplay terminates when the user either reaches an edge (correct or incorrect) or if the user fails to reach an edge within 60 seconds. For each trial, the program records whether the user was successful or unsuccessful, in addition to the EEG power features at each step.

After the user plays several trials, we utilize a linear regression algorithm to readjust the weights for each of the features. That is, we attempt to find weights for power features that distinguish attempts to move in one direction or another. It is then possible to either continue play of the training game and update weights again (this time over all trials encountered thus far), or to continue to the third stage of training.

Third and finally, the user plays a training game where the user attempts to move the object to the corner of the screen, rather than the edge. These attempts can be interpreted as ground-truth intent to move in two directions; for example, movements when the goal is the northwest corner can be interpreted as attempts to move up and left simultaneously. The purpose of this stage of the procedure is to give the user practice in controlling multiple dimensions simultaneously. Once again, we record the result and data from the EEG signal, and we update the weights of the algorithm periodically using linear regression.

4.3 Gaze Tracking

Blink Detection One function of the gaze tracking system is to detect whether a blink has occurred. Since a blink renders the EEG data unreliable, gaze tracking can cancel any potential movement that may have been classified by the gaze tracking system. The Fove API has a

function that checks whether or not a user’s eye is closed. In case the user’s eye was closed at any point during the previous third of a second, a blink is detected, and any motor commands are ignored. A script in Unity consistently monitors whether an eye is closed in order to make this decision.

Object Localization The second, and perhaps more central function of the gaze tracking system is to determine whether the user is looking at an object. If the user is not presently gazing at an object (as defined by whether or not the location gazed at coincides with the pixel display, then no updates to the virtual reality game will commence. However, if the user is looking at an object, and no blink has been detected in the past 1/3 second, then the object moves according to the command. For some objects, a Fove plugin allows some objects to determine if the user is looking at them. This calculation is made by taking into account both the user’s head position and gaze vector. A second script, which monitors if any objects are looked at, moves the selected object in the direction specified by the EEG.

5 Implementation

Figure 4 describes the implementation of the final, trained application. The core functions are implemented with Node.js and Unity, since these languages support API calls to the EPOC and Fove headsets, respectively. Unity scripts are responsible for receiving gaze tracking information from the headset, as well as processing if an object is being viewed and is commanded to move. However, the code that calculates the Fourier transform from the raw EEG time series data is an exception; here, Node.js calls a Python function and uses its results to perform classification.

This project conducted relatively less programming in Unity than in Node.js. However, one tweak in Unity for the sake of object localization is worth mentioning. The original object in Unity that moved when gazed at moved in only one dimension, depending on whether the left eye or right eye (or both) were looking at

it. For the purposes of this project, we modified this section of the code to allow the object to move in two dimensions and according to the detected command for direction. Retrieving this command in Unity required communication from the Node.js program.

However, communication between these languages raises a problem, since it is not currently possible to call functions of one language from the other. To amend this issue, we implement a simple file buffer that contains the most recent instructions to move: the Node.js program constantly re-writes the file, and the Unity program reads the file whenever it is necessary to move. While there may be a more elegant way to transmit this information, this file buffer system communicates the necessary bits of information while minimizing the amount of information that relies on this communication.

Experiments concerning both gaze-tracking and EEG data were conducted on a Dell Latitude laptop with operating system Windows 10. As of writing, Windows is the only operating system that the Fove headset supports. The Fove headset itself required two USB connections and a HDMI port. A position camera, included with the Fove headset, tracked head movements (in order to adjust the display to mimic visualizing a 3D environment) and required an additional USB port.

The Emotiv EPOC communicated with the Dell laptop via a wireless connection to a USB dongle. The Cortex API, which allowed Node.js to access data from the EEG, required a subscription to an Emotiv service in order to enable retrieval of raw data. Therefore, it was required that the user provide credentials for a subscribing account both in the Node.js code retrieving data and in CortexUI, an Emotiv application running in the background. While initial experiments with the EPOC, including some initial training of one and two-dimensional control, were run on a Macbook Pro laptop, later experiments shifted to the Dell laptop in order to accommodate gaze tracking.

The elastic bands of the Fove device were worn on top of the EPOC, which utilized a rigid plastic structure to place sensors. Wearing the

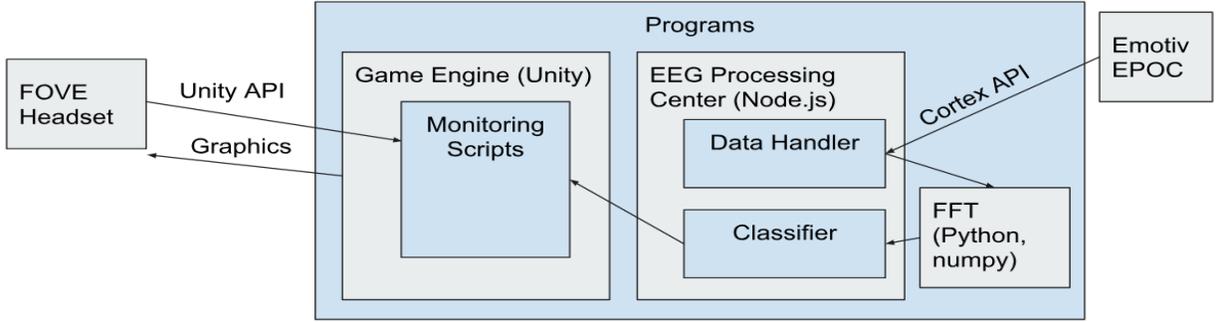


Figure 4: Implementation flow of the program.

Fove device did not result in a need to adjust the position of EPOC sensors. CortexUI provided instructions on how to properly wear the EPOC so that the sensors are in their standard position. The felt sensors required dampening via a saline solution in order to achieve optimal recording quality. CortexUI also provided a color-coded map of sensors indicating the quality of each sensor’s contact with the scalp. Experiments were only conducted after each sensor was colored green, indicating a good quality connection.

6 Results

We have presented a framework for how the user can learn to improve her proficiency in controlling the BCI via a procedure of automatic weight adjustment and tuning. It remains to show whether this training procedure actually is capable of improving system control. While this is difficult to analyze during online gameplay, since we lack a way of defining the “ground truth” movement a user intends at every step, we can still evaluate the reliability of the system based on recordings of earlier training sessions. The data from these sessions can yield insight into how much the user’s ability to control mu and beta rhythms has developed.

To evaluate the effectiveness of training, we evaluate the ability of a linear regression model to accurately predict either vertical or horizontal commands from the mu and beta rhythm features collected during a prior training session. If

the linear regression model performs better, then arguably the user has learned to create better features via more proficient command of the mu or beta rhythm. Since the user was given the explicit goal of attempting to move in a certain direction, we have ground truth knowledge of what they were aiming for, allowing us to train a supervised learning algorithm to model the data.

Therefore, we evaluate two datasets: “Early EEG” and “Late EEG.” These datasets in turn have two components: vertical movements and horizontal ones. The mu and beta features collected during Early EEG occurred without prior training with the Emotiv EPOC or the training program. That is, the user was told by a script to think of a certain command without receiving feedback on whether it was recognized as successful. By contrast, Late EEG features were collected after training with the one-dimensional and two-dimensional training interfaces implemented in Node.js. Each dataset contains a balanced number of vertical and horizontal movements; there are 408 sets of features for up/down movement and 424 sets of features for left/right movement. (Each set of features is a vector of 4 beta or mu features, depending on the dimension evaluated.)

Both datasets were the result of recordings for a single user. The “Early EEG” dataset utilized a shell script that asked the user to imagine moving left, right, up, or down via a text prompt on the console display. A round consists of asking the user to perform each of these imagination tasks once. After each prompt, the user is given 3 seconds to prepare before the program

starts recording raw EEG data. Raw EEG signal is saved to a CSV file, though for purposes of comparison, we convert these signals into the beta/mu features used later on. The program ends after 3 seconds, from which we can collect 8 sets of beta/mu features. In total, we conducted 60 rounds, although due to a connection error with the online connection to the Emotiv service, some of these rounds resulted in a corrupted recording. We therefore had to remove the corrupted recordings, which slightly reduced the total amount of data available. (If no data were corrupted, we would have 480 sets of features.)

The "Late EEG" dataset was collected from user data in playing a one-dimensional training game. The program directly saves the mu/beta features at each time step to a CSV file. The training game in the horizontal dimension lasted for approximately 12 seconds, and the training game in the vertical dimension lasted for approximately 6 seconds. The difference in the duration of the games is an artifact of the oblong size of the terminal the training game is played in: the user's goal is to make it to the edge of the screen within a limited amount of time, and the default terminal size has greater width than height. For this reason, we collected a higher number of vertical trials than horizontal trials. In total, we recorded 16 trials of moving left, 16 trials of moving right, 27 trials of moving down, and 28 trials of moving up. Horizontal trials produced 35 sets of beta/mu features each, and vertical trials produced 16 sets of beta/mu features each. Since these produce more sets of features than in the earlier dataset, we limit the dataset to only more recently recorded trials in order to make sure the size of the datasets are the same. This limitation ensures a fair comparison of the quality of the data, since quantity of data available would likely affect the performance of the classifier.

To evaluate the quality of the linear regression model, we partition the dataset into two components: training (80%) and validation (20%). Since the model is trained only on the training data, the validation error is a helpful indicator of how well the model generalizes to previously unseen data. For each data set, we

perform 50 random partitions and compute the average validation and training accuracy across the partitions. Performing multiple random partitions allows us to compute a confidence interval for each accuracy. The following table lists the results, with the higher percentages for each type of set in bold.

Dataset type	Early EEG	Late EEG
Horizontal (training)	52.10 \pm 2.05	58.75 \pm 0.92
Horizontal (validation)	47.99 \pm 3.12	57.76 \pm 3.47
Vertical (training)	53.56 \pm 1.06	58.54 \pm 1.14
Vertical (validation)	51.28 \pm 3.29	57.01 \pm 3.60

Two notable results arise from this table. First, and perhaps most notably, the validation accuracy for both horizontal and vertical movements in the Late EEG is significantly better than chance (50 percent). Since the validation accuracy is not better than chance in the Early EEG data, it is reasonable to suggest that the training protocol helped contribute to the performance increase. Second, the mean accuracy values for the Late EEG are higher than those of the Early EEG for every comparable category, also pointing to an increased ability of the user to generate helpful features. In particular, the mean validation accuracy increases by almost 10 percent and almost 6 percent in the horizontal and vertical cases, respectively.

We claim that in context, this increase in validation accuracy demonstrates the promise of the training procedure previously outlined. While it is unclear how well a user could possibly control the Emotiv EPOC, we have already seen how the device fares its disadvantages due to its relatively lower sampling rate and sub-optimal sensor locations for this task. In spite of these disadvantages, however, we have shown that these disadvantages are not insurmountable, since it is possible for the user to improve performance.

There may be other optimizations to achieve even greater accuracy independent of requiring the user to undergo training, such as increasing the quantity of data trained on or by using a

more sophisticated classification algorithm. The purpose of this comparison with linear regression is to show that it is possible for a user to improve performance even with a very simple learning algorithm. This is presumably done by the user learning to generate better brain wave signals that a program can decode. Thus, the training procedure developed here can be seen as complementary to other tactics to improve performance.

7 Related Work

Sensorimotor BCI. [15] demonstrated proof of concept that users could control a sensorimotor BCI with control in one dimension after several hours of training. Research in the lab since then, as described in findings such as [14], [16], [13], [17], [21], and [22], have improved on expanding the functionality of the interface, increasing the number of dimensions users could control and developing techniques to automatically, rather than manually, set the parameters for the interface. This thesis utilizes methods in sensorimotor BCIs developed in these papers, including using machine learning to learn good parameters for user control and using the power of frequencies as features. The additional contribution of this thesis lies in first, adapting these techniques to work for a commercial-grade headset with significantly lower spatial resolution (14 vs. 64 electrodes) and second, in augmenting the sensorimotor BCI with gaze tracking to allow for control of multiple objects.

BCIs using Emotiv EPOC. Several studies have attempted to decode signals from the Emotiv EPOC, for various tasks such as color imagination [23], cognitive workload [18], and motor imagery [19]. Although [19] showed that motor imagery detection was possible on the Emotiv EPOC, they merely analyzed classification results on a recorded dataset, rather than during live game play. Other studies, such as [20], focused on using the Emotiv EPOC to control a simple Bloons tower defense game. [20] used a SSVEP-based BCI, which used the BCI to determine what a user was looking at. While this example is one of the earliest to demon-

strate a workable BCI on the Emotiv EPOC, the rate of user control was determined by the rate the visual stimuli corresponding to the SSVEP changed, making it difficult for the user to rapidly transmit commands.

Gaze tracking and BCI. While no previous projects have used a BCI and gaze tracking for multi-object control, a few have combined the technology for different purposes. [6] combine a SSVEP-based BCI and gaze tracking to reach greater precision for a BCI enabling typing. [11] presented an early example of a system that combined EEG and gaze tracking for the purposes of object selection in three dimensions. Gaze tracking data determined where on the 2D screen the user was looking, and imagined arm movements determined where in the depth dimension to select. While this result was not implemented as a brain-computer interface, it served to show that integration of this information is possible. [8] is perhaps the most similar project to our approach, combining an Emotiv EPOC device with eye tracking in order to control a quadcopter. However, the approach allowed for only control of one object, since the BCI was used to determine user concentration and the eye tracking selected a direction to move. By contrast, this project investigates control of multiple objects, each of which can move in two dimensions.

8 Conclusion

We have presented a portable, non-invasive brain-computer interface that runs on commercial-grade hardware, allows for control of multiple objects in virtual reality and is supported by a training framework. Results indicate the training framework does indeed allow for better control of the interface. The program performs online decoding of EEG signals and maintains checks against possible sources of interference, such as blinking. To our knowledge, this is the first non-invasive brain-computer interface to allow control of multiple objects in multiple dimensions. Nevertheless, the project may benefit from the following extensions:

- Addition of a neutral command. This could be done in multiple ways, for example by developing a three-way classification procedure when training the user with a brain-computer interface or by detecting the level of user concentration alongside command detection and failing to move if levels of concentration are not high enough.
- Comparison to alternate brain-computer interfaces with the same equipment. A reviewer pointed out that the same hardware could be mobilized to accomplish the same task of multi-object control in a very different way. Perhaps gaze motion, extracted from gaze positions, could be used to signal object motion and SSVEP-based BCI could be used to determine the object. It may then be possible to compare which system is preferred by users for multi-object control.
- Building practical learning applications to apply the system toward. One particular approach that seems promising is to develop a game that requires multitasking and proficient control of at least 2 objects in a given scene. [1] found that repeated play of a clinically designed multitasking game improved cognitive control and even attention, especially for aging populations. Since BCIs have the potential for transmitting a high amount of information each second, they could enable novel multitasking games that could also reap cognitive benefits.

References

- [1] Joaquin A Anguera, Jacqueline Boccanfuso, James L Rintoul, Omar Al-Hashimi, Farhoud Faraji, Jacqueline Janowich, Eric Kong, Yudy Larraburo, Christine Rolle, Eric Johnston, et al. Video game training enhances cognitive control in older adults. *Nature*, 501(7465):97, 2013.
- [2] Mark F. Bear, Barry W. Connors, and Michael A. Paradiso. *Neuroscience: exploring the brain*. Wolters Kluwer, 2015.
- [3] Ujwal Chaudhary, Bin Xia, Stefano Silvoni, Leonardo G Cohen, and Niels Birbaumer. Brain-computer interface-based communication in the completely locked-in state. *PLoS biology*, 15(1):e1002593, 2017.
- [4] Stefan Ehrlich, Ana Alves-Pinto, Renée Lampe, and Gordon Cheng. A simple and practical sensorimotor eeg device for recording in patients with special needs. In *Neurotechnix2017, CogNeuroEng 2017*, 2017.
- [5] Reza Fazel-Rezai, Brendan Z Allison, Christoph Guger, Eric W Sellers, Sonja C Kleih, and Andrea Kübler. P300 brain computer interface: current challenges and emerging trends. *Frontiers in neuroengineering*, 5:14, 2012.
- [6] Felix Gemblar, Piotr Stawicki, and Ivan Volosyak. A comparison of ssvep-based bci-performance between different age groups. In *International Work-Conference on Artificial Neural Networks*, pages 71–77. Springer, 2015.
- [7] Christian Herff and Tanja Schultz. Automatic speech recognition from neural signals: a focused review. *Frontiers in neuroscience*, 10:429, 2016.
- [8] Byung Hyung Kim, Minho Kim, and Sungho Jo. Quadcopter flight control using a low-cost hybrid interface with eeg-based classification and eye tracking. *Computers in biology and medicine*, 51:82–92, 2014.
- [9] Matt Lang. Investigating the emotiv epic for cognitive control in limited training time. 2012.
- [10] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436, 2015.
- [11] Eui Chul Lee, Jin Cheol Woo, Jong Hwa Kim, Mincheol Whang, and Kang Ryoung

- Park. A brain-computer interface method combined with eye tracking for 3d interaction. *Journal of neuroscience methods*, 190(2):289–298, 2010.
- [12] Sergio Machado, Oscar Arias-Carrion, Flávia Paes, Pedro Ribeiro, Mauricio Cagy, Roberto Piedade, Leonardo Ferreira Almada, Renato Anghinah, Luis Basile, Maria Francesca Moro, et al. Changes in cortical activity during real and imagined movements: an erp study. *Clinical practice and epidemiology in mental health: CP & EMH*, 9:196, 2013.
- [13] Dennis J McFarland, Dean J Krusienski, and Jonathan R Wolpaw. Brain-computer interface signal processing at the wadsworth center: mu and sensorimotor beta rhythms. *Progress in brain research*, 159:411–419, 2006.
- [14] Dennis J McFarland, A Todd Lefkowitz, and Jonathan R Wolpaw. Design and operation of an eeg-based brain-computer interface with digital signal processing technology. *Behavior Research Methods, Instruments, & Computers*, 29(3):337–345, 1997.
- [15] Dennis J McFarland, Gregory W Neat, Richard F Read, and Jonathan R Wolpaw. An eeg-based method for graded cursor control. *Psychobiology*, 21(1):77–81, 1993.
- [16] Dennis J McFarland and Jonathan R Wolpaw. Sensorimotor rhythm-based brain-computer interface (bci): feature selection by regression improves performance. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 13(3):372–379, 2005.
- [17] Dennis J McFarland and Jonathan R Wolpaw. Sensorimotor rhythm-based brain-computer interface (bci): model order selection for autoregressive spectral analysis. *Journal of neural engineering*, 5(2):155, 2008.
- [18] Timothy McMahan, Ian Parberry, and Thomas D Parsons. Modality specific assessment of video game player’s experience using the emotiv. *Entertainment Computing*, 7:1–6, 2015.
- [19] T Mladenov, K Kim, and S Nooshabadi. Accurate motor imagery based dry electrode brain-computer interface system for consumer applications. In *Consumer Electronics (ISCE), 2012 IEEE 16th International Symposium on*, pages 1–4. IEEE, 2012.
- [20] Marijn van Vliet, Arne Robben, Nikolay Chumerin, Nikolay V Manyakov, Adrien Combaz, and Marc M Van Hulle. Designing a brain-computer interface controlled video-game using consumer grade eeg hardware. In *Biosignals and Biorobotics Conference (BRC), 2012 ISSNIP*, pages 1–6. IEEE, 2012.
- [21] Jonathan R Wolpaw, Niels Birbaumer, Dennis J McFarland, Gert Pfurtscheller, and Theresa M Vaughan. Brain-computer interfaces for communication and control. *Clinical neurophysiology*, 113(6):767–791, 2002.
- [22] Jonathan R Wolpaw and Dennis J McFarland. Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans. *Proceedings of the National Academy of Sciences of the United States of America*, 101(51):17849–17854, 2004.
- [23] Je-Hun Yu and Kwee-Bo Sim. Classification of color imagination using emotiv epoc and event-related potential in electroencephalogram. *Optik-International Journal for Light and Electron Optics*, 127(20):9711–9718, 2016.