Towards a Wearable Device for Controlling a Smartphone with Eye Winks

Davide Valeriani
Brain-Computer Interfaces Laboratory
School of Computer Science and Electronic Engineering
University of Essex
Colchester CO4 3SQ, UK
Email: dvaler@essex.ac.uk

Ana Matran-Fernandez
Brain-Computer Interfaces Laboratory
School of Computer Science and Electronic Engineering
University of Essex
Colchester CO4 3SQ, UK
Email: amatra@essex.ac.uk

***Abstract*—The development of mobile technology over the last years and the consequent boom of available apps has enabled users to migrate a wide range of activities that were traditionally performed on computers to their smartphones. Despite this new freedom to work ubiquitously, there are circumstances in which operating the device becomes difficult, e.g., when the hands are not free due to driving or other activities. Even though there are voice-control alternatives for operating smartphones, these do not perform well in crowded or noisy environments. In this paper we present *EyeWink*: an innovative hands- and voice-free wearable device that allows users to operate the smartphones with eye winks. The system records the Electrooculography (EOG) signals on the forehead by means of two facial electrodes. Eye winks are detected by comparing the potentials recorded from the electrodes, which also helps avoid false actuations due to (unavoidable) eye blinks. The user can associate the action to perform with each eye by means of an app installed on the smartphone. The proposed device can be widely used, with customers ranging from runners to people with severe disabilities.**

I. INTRODUCTION

Smartphones have become a pervasive tool in our everyday lives. More than 60% of adults in the United Kingdom own a smartphone, and a similar percentage holds in many countries over the world [1]. The improvements in performance due to more powerful hardware and the integration of a broad range of sensors, such as accelerometers and cameras, have allowed an increased number of applications and uses, including education [2] and healthcare [3]–[5].

In recent years, smartphones have also been broadly used in human-computer interaction to enhance the user's experience for controlling a large range of devices and applications. Various studies have shown how these devices could be used to help people with impairments in their everyday lives [6], [7]. Moreover, alternative input modalities, such as those based on voice [8] or eye movements [9], have been developed to operate the smartphones without hands, allowing users to perform multiple actions at a time.

Voice-controlled smartphones usually require quiet environments to work properly. Hence, they cannot be reliably operated while performing outdoor activities or when other people are around (i.e., in noisy environments). Moreover, recent research has shown that many people prefer to avoid using voice activated personal assistants, such as Siri [10] or Google Now [11], in public spaces due to privacy concerns [12]. Finally, the use of voice to control a smartphone or other device causes an increase on the mental workload of the user and, therefore, a reduction in his/her level of attention [13]. Hence, this alternative input channel cannot be used in contexts where an high level of attention is required, e.g., in driving.

On the other hand, current systems that allow the user to control a device with the eyes are mostly based on eye tracking [14], [15] and, therefore, would require the smartphone to be placed in front of the user. This significantly reduces their potential use in everyday life (e.g., they cannot be used in the darkness, or while using sunglasses) and their contribution to hands-free technologies, since people would require to hold the smartphone in front of them to be able to use it. Eye tracking can be done using the smartphone's camera, but there are other options, such as Tobii [16], that also allow the user to control computers and tablets. However, the high cost of these devices significantly restricts their use by disabled people.

In this paper, we propose an innovative and low-cost wearable device to control the smartphone (and, in principle, any device) with eye winks. The system records muscular potentials by means of two electrodes placed either above or laterally to each eye to detect winks and filter out the blinks. Left and right winks are then transformed into commands and streamed to the smartphone via Bluetooth. The commands received are mapped into specific actions through an app that is installed on the smartphone.

The paper is organised as follows: Section II describes the architecture of the system, from both the hardware and the software perspectives. Section III presents and analyses results obtained with a pilot study. Finally, in Section IV we conclude the paper, drawing some conclusions and pointing out the main challenges to bring this device to the market.

II. METHODS

This section describes the various parts that comprise *EyeWink*. There are two main constituents of the system: the hardware component and the software, which in turn performs data processing and executes the actions desired by the user. A schematic of the system can be seen in Fig. 1, which we will refer the reader to throughout the rest of this section.

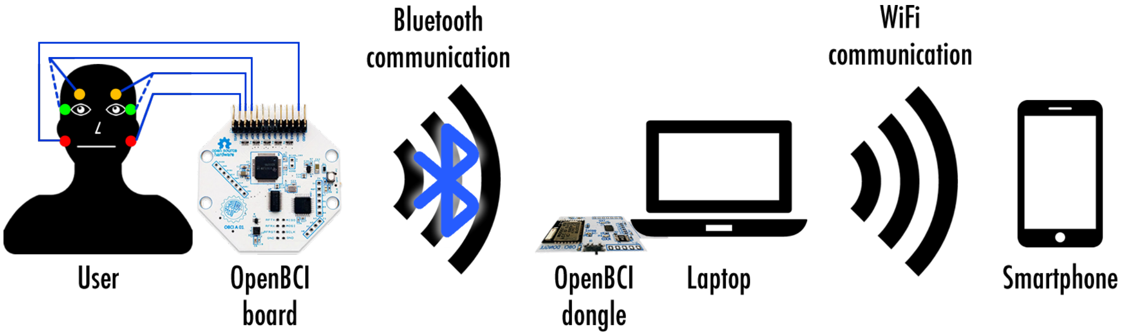


Fig. 1: Architecture of the system. Active potentials are registered by means of two electrodes placed above or distally with respect to each eye. The signals are then amplified by an OpenBCI board and sent to a computer via Bluetooth. Eye winks are extracted after processing and the commands are then sent to the smartphone via WiFi.

A. Data Acquisition

Two active electrodes are placed on the user's face, either on the forehead (about 3 cm above each eye and horizontally aligned to the center of the pupil when the user is looking ahead — highlighted in orange in Fig. 1), or distally, one on each side of the face (on the temporal bones, above the zygomatic bones and aligned vertically with the center of the pupil — depicted in green in Fig. 1). In addition to these, the system needs one mastoid reference electrode behind each ear (represented in red in the figure). All the electrodes are securely placed on the skin by means of medical tape.

Active potentials representing muscular activity (Electromyography — EMG) and, in particular, Electrooculography (EOG, i.e., voltage fluctuations associated with eye and eyelid movements) are recorded and streamed for processing by means of an 8-bit OpenBCI board. Each EOG channel is referenced to the corresponding mastoid electrode. Data are sampled at a frequency of 250 Hz and sent to a laptop for processing (more on this below).

B. Data Processing

Biosignals are inherently contaminated by artefacts (i.e., noise). Even though the electrodes from the prototype were secured to the skin, small displacements due to movements in the regions where they are placed (especially those used to record EOG data) will affect the signals and create drifts in the recorded potentials. For these reasons amongst others, the signals need to be processed before the system can determine whether the user is trying to send a command to the device. Data received through the OpenBCI dongle (see Section II-D) are accumulated into 100-sample-long "epochs" (i.e., a finite sequence of data) before they can be processed. At the given sampling rate of 250 Hz, we are effectively processing blocks of data that are 400 ms long.

The incoming data is re-referenced through a Common Average Reference (CAR), which is used to remove sources of background noise. Put simply, the CAR uses the mean of all electrodes at a given time sample as a reference value for background noise (which is assumed to be equally present across all electrodes), and subtracts this value from each sample of the epoch.

We then use a 0.15-40 Hz Butterworth band-pass filter of order 4 to remove noise from the signals, such as mains noise and EMG (typically present at frequency ranges >60 Hz) [17]. The processed epochs are then fed to the decision-making part of the system, which is described in Section II-C.

The processing is very limited and can be done in real time before sending the output command to the smartphone. Moreover, the reader should remember that the system described here is just a prototype. In the future, the data recording hardware should be able to transmit the data wirelessly to the smartphone, where the processing stage will have to be implemented. Despite the increase in processing capabilities of smartphones over the last years, one should still try to keep the load at a minimum (while ensuring good performance of the system). The system described here can be easily migrated to the phone without significantly impacting battery life.

C. Wink Detection and Blink Removal

With *EyeWink*, the user is able to send commands to an external device (i.e., a smartphone) by means of eye winks. These are easily produced by the user at will. However, a person cannot stop blinking and, so, it is very important to be able to distinguish the blinks from the winks so that only the latter affect and control the behaviour of the smartphone.

Our system is capable of distinguishing between involuntary blinks and voluntary left/right-eye winks by comparing the peak amplitudes recorded by either electrode. In particular, processed epochs are passed to a peak detector, which calculates, for each eye separately, the maximum value recorded in the epoch (A_l and A_r, for left and right eye, respectively), as well as the difference between these two values ($D = A_l - A_r$).

The maxima are first compared to a fixed threshold T_1. This step is used to determine whether the epoch contained a blink or wink (which are represented by large, rapid voltage fluctuations). Both eye winks and eye blinks will have an influence on the two facial electrodes, due to the conductance properties of the skin and, depending on the position of the electrodes, on the distances between them and to the opposite eye. Because of this, a small blink-like potential will be recorded in the opposite eye to that of the one that the user winked (more on this on Section III). This wave will be much smaller in the eye that remained open, so we used this

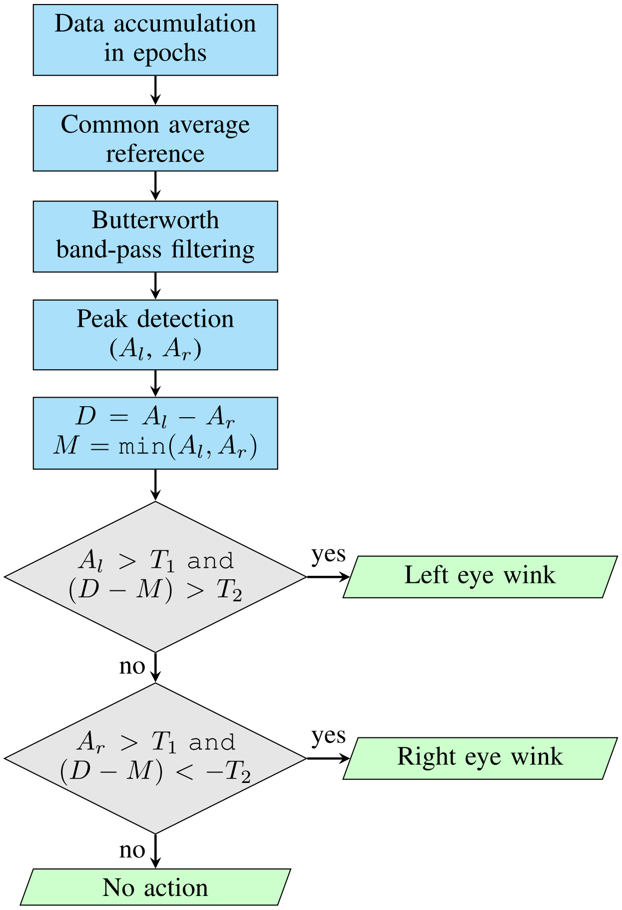


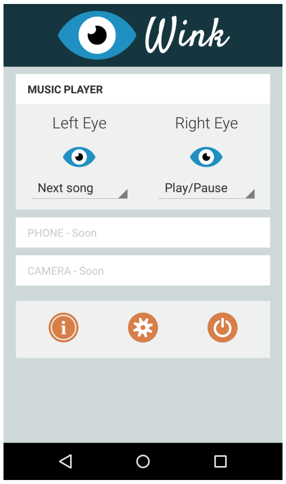
Fig. 2: Flowchart of the algorithm used for data processing and eye winks classification.

property of EOG signals to train our system to distinguish between blinks and winks by comparing A_l and A_r vs D. If $D - \mathtt{min}(A_l, A_r) > \mathtt{abs}(T_2)$, then there was an eye wink.

The entire algorithm is described in Fig. 2. Given the simplicity of this method, *EyeWink* can be used without the need for long training sessions. In the prototype, the thresholds T_1 and T_2 are hardcoded in the program that processes the data and sends the command to the smartphone. This requires the experimenter to ask the user to wink 4–5 times with each eye so that the thresholds can be adjusted properly. These thresholds usually need to be adjusted depending on the posture of the participant (i.e., sitting, standing, walking), since there is a higher influence of muscle artefacts when the person is standing/walking. However, once the processing is merged in the app, the user will have a way of adapting these thresholds to the "strength" of his/her eye winks, also offering the possibility of fixing different thresholds for each eye (already implemented, see Section II-E).

D. Communication

The OpenBCI board can communicate wirelessly to other devices using the Bluetooth Low Energy (BLE) protocol. However, the current firmware does not support direct BLE connections to smartphones or tablets [18]. Therefore, in this prototype we used the provided USB dongle to receive the data from the board through a BLE connection.



Left Eye: Next song

Right Eye: Play/Pause

Fig. 3: Main interface of the Android app used to configure the actions on the smartphone. The top panel allows the user to select the actions that should be associated with each eye wink. The lower panel contains three buttons providing (from left to right) further information about the app, a window to change the settings and the button to enable/disable the device, respectively.

The dongle streams the received data over the serial port of the host (i.e., laptop), which creates the epochs described in II-B and processes them to extract eye winks and convert them into commands (as described above). The resulting value is then sent to the smartphone via a TCP socket over a previously established WiFi connection.

E. Smartphone App

In order to allow the user to decide which action he/she would like to perform when winking with the left or right eye, we developed a smartphone app. At the moment, it is currently supported by Android systems only. The main interface of the app is shown in Fig. 3.

The current prototype supports the following commands to operate the music player on the smartphone:

- "Play/Pause", that allows to play the music (or stop it if it is already playing);
- "Next song", that plays the next song in the current playlist;
- "Prev song", that plays the previous song in the current playlist or rewinds the current one;
- "Volume up", that increases the volume by a 10% factor;
- "Volume down", that decreases the volume by a 10% factor.

The user can select which action should be associated to each eye wink by picking a command from the two dropdown menus in the app. A "no action" (or "disable") option is also available in case the user wants to disable the device for one or both eyes.

The *EyeWink* app also provides a connect button (i.e., bottom right button in Fig. 3) that allows the user to enable or disable the connection to the device. When the button is pressed for the first time, the app connects to the TCP socket opened by the laptop to read the commands (in the prototype version). The connection is managed by an Android service, that allows to keep the connection opened even if the user sends the app to the background. To disconnect from the device, the user has to press the connect button again (or exit the app).

The settings button (i.e., bottom middle button in Fig. 3) allows the user to set the parameters required by the app to connect to the TCP socket (i.e., hostname and port). Moreover, it provides a slider to adjust the thresholds used to extract eye winks and, thus, the sensitivity of the device to the strength of the winks[1].

Despite the fact that the app of the prototype only allows to control the music player, extra controls or actions (such as answering a call or taking a picture) can be easily implemented, as this only requires to associate the value received from the TCP socket to a specific action on the smartphone.

III. RESULTS

The system has been tested on two users (mean age 28, one female, one left-handed) using the device on different days and running a *cue-based* experiment and a *free-mode* one.

The accuracy of our device at detecting eye winks has been measured by using the F_1 score given by

$$F_1 = \frac{2 \cdot TP}{2 \cdot TP + FN + FP}$$

where TP (true positive) is the number of correct commands sent to the device, FN (false negative) is the number of times that the device did not detect the wink and FP (false positive) is the number of false alarms (i.e., the number of times when the device executed a command even though the user was not trying to send one).

A. EOG analysis

We start our analysis by looking at the recorded signals representing eye winks, eye blinks and idle state, in order to evaluate the performance of the preprocessing. Representative plots of these are shown in Fig. 4.

As can be seen, both eye winks and blinks are characterised by a drop in voltage. However, in the case of an eye blink (Fig. 4(top right)) the drop is registered across both electrodes, which allows our algorithm to filter them out by considering the difference in voltages between the left and the right eyes.

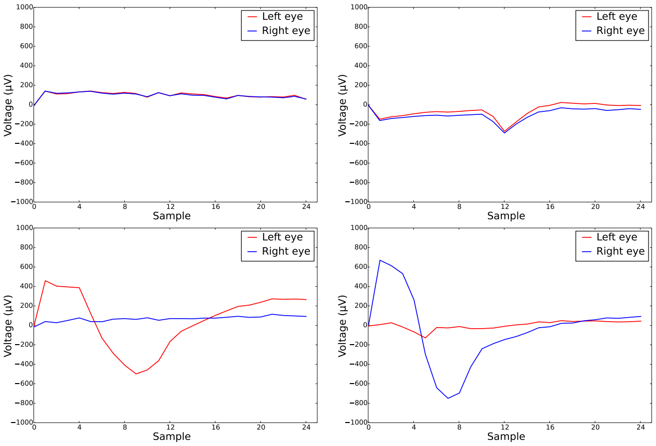


Fig. 4: Representative plots of idle state (top left), eye blink (top right), left eye wink (bottom left) and right eye wink (bottom right). Data were collected from one participant sitting on a chair with the electrodes placed on the forehead and subsampled by a factor of 10.

On the other hand, eye winks, as depicted in Fig. 4(bottom), show a pronounced positive peak followed by a negative one in the eye that is being winked. This sudden increase and drop is much larger in amplitude than that of the eye blinks (the scale used for the ordinate axis across all plots in Fig. 4 is the same). Furthermore, the amplitude in the "resting eye" remains almost unchanged with respect to the baseline, so our algorithm is capable of discerning that the user was trying to send a command.

B. Cue-based performance

In the cue-based experiment, the two participants first sat on a chair and were asked to play the music with a left eye wink and to change the song by winking with the right eye. Each participant started the experiment by performing a set of 4 winks with each eye in order to calibrate the device (i.e., adjust the thresholds for wink detection). Then, they performed multiple sequences of left and right winks for a total of 20 of each type. In addition to this, we included 20 trials that were free of winks and blinks (i.e., idle state), and an extra 20 trials with blinks. The order in which these were presented was randomised and told to the volunteer by means of a screen in which the words "blink", "right wink", "left wink" and "do nothing" were displayed sequentially.

The same protocol was repeated with the participants standing in front of a window and walking around the room, without following any specific path (repeating the calibration in between operation modes, but not within each of them).

We also compared the performance of the device when the electrodes were placed either on the forehead or distally, as described in Section II-A.

The results are shown in Table I. As can be seen, the device is more stable when operated by a user sitting on a chair. This is reasonable as, in this condition, most of the muscles are resting and, therefore, the impact of the muscular artefacts on the system is negligible. However, when the users were standing or walking, the performance drops slightly, which is the reason why we readjusted the thresholds before starting on a new operation mode.

---

[1] As we said above, this part of the system can be configured, even though for the prototype described in this paper the processing was all done on the computer, so the user adjusting the thresholds on the phone had no effect on the behaviour of the device

TABLE I: F_1 scores of the device in multiple configurations.

Configuration	Participant A		Participant B	
	Forehead	Distal	Forehead	Distal
Sitting	0.90	0.91	0.85	0.84
Standing	0.83	0.85	0.79	0.78
Walking	0.80	0.83	0.75	0.73

On the other hand, the position of the electrodes does not seem to have an impact on the performance. Both participants were able to easily readjust the sensitivity of the device through the app to make it work with the new electrodes' location. Higher sensitivity (i.e., thresholds T_1 and T_2) was required when the electrodes were placed distally, since this configuration is slightly more prone to false positives due to artefact contamination from the muscles around the mouth.

C. Free-mode performance

When trying to measure the performance of an online system in which the user can freely operate at will, experimenters need to rely on either self reports or annotations (e.g., by recording the user while he/she uses the system and then associate the actions that are inferred from the behaviour of the user to the actual behaviour registered in the phone), or both.

Since one of the goals of the experiment was to determine whether *EyeWink* could be used in a realistic environment in which the user moves freely and talks, we asked one of the volunteers to return for more sessions of "free will" use, in which she could operate the music player with eye winks at her will. This test was done in a public park and she was free to move or to sit (as long as she remained within Bluetooth range distance of the laptop, which was placed on a bench).

Performance was evaluated by the experimenter by counting how many times the command executed on the phone (play/pause music or change the song, as in the previous test) corresponded to the action performed by the participant (blink, left/right wink, idle).

The calibration of the system (threshold adjustment) was done with the subject standing, and the facial electrodes were place in the distal position (this was her choice when asked where did she prefer them). She was also told that she could use the system for as long as she wanted.

The system achieved an overall F_1 score of 0.85 in the 2 hours that the experiment lasted. During this time, the volunteer walked, talked to her friends and was even able to eat while wearing the device without an increase in the rate of false positives. At times she sat down and the system was still able to detect the winks and filter out the artefacts from blinks and other sources of EMG activity without readjusting the threshold.

At the end of the experiment, she mentioned that she had really enjoyed the experience and, for a while, she actually kept trying to control the music player on her own smartphone via winks.

IV. CONCLUSIONS AND FUTURE WORK

In this paper we presented an algorithm to detect eye winks and filter out blinks with two facial electrodes. We used this algorithm to interpret and classify EOG signals from two volunteers and allow them to control a smartphone hands-free and without the need of having the screen in front of them by using a prototype operating on an OpenBCI board.

We tested two different configurations for the facial electrodes (on the forehead and distally) and found no differences in performance between them across the three modes of operation (sitting, standing and free movement). This will allow us to install the electrodes on multiple supports, such as a headband, cap or glasses, making the transition of the system towards a wearable device easier. This is one of the main challenges that we will tackle in the future: developing the hardware (wearable device plus transmission board to stream the data) whilst keeping the device affordable for prospective users. In particular, the electrodes that we used for the prototype need gel to decrease the impedance between the electrode and the skin in order to obtain a clean signal (i.e., we used wet electrodes). This is not practical, and in the future we will work on developing dry electrodes which can offer the same (or similar) resolution.

Furthermore, we found no significant differences across the three modes of operation within each participant, which suggests that the design is robust to motion artefacts and allows the device to be used in situations when the user is on the move, e.g., while practising sports such as running. In particular, the most common error in the system was sending one command twice after the user had winked. This is probably due to the epochs being too short and the wink overlapping over consecutive epochs. However, we believe that the length of the epoch (400 ms) represents a good compromise between performance and the time that it takes for the system to recognise the command and execute it (longer epochs would mean that the user needs to wait to see if the command was successfully sent).

Another group of users that might be interested in a device such as *EyeWink* is the disabled community. Brain–computer interfaces allow users to control external devices by means of electroencephalography signals. In this field, EOG artefacts are considered noise. However, when a severely paralysed user still retains control of their eyes, other alternatives (generally by means of eye trackers, such as the aforementioned Tobii) are available. *EyeWink* provides a reliable and affordable alternative to such eye-tracking systems. For example, the app on the smartphone could be easily expanded to allow users to control a keyboard on the screen by using winks to move through it and spell words.

In the future, we will further develop *EyeWink* to bypass the computer and stream the data via Bluetooth directly from the board to the smartphone, in order to increase the freedom of movement for the user. The data processing required for EyeWink can easily run on a smartphone without performance issues. In particular, the main obstacle for this improvement would be the implementation of a firmware to hold the communication.

Finally, the current version of the *EyeWink* app can only be used to control the music player of the smartphone. In order to properly control the phone, we need to expand the number of commands that the user can send (e.g., by supporting combinations of eye winks and blinks), as well as the amount of actions that these can be mapped to (e.g., by allowing users to control a keyboard, or accept/reject phone calls). Moreover, at the moment the app is is only supported by Android systems, but we will also expand it to iOS in the near future.

REFERENCES

[1] Ofcom, *International Communications Market Report*, 2014.

[2] B. D. Honegger and C. Neff, "Personal smartphones in primary school: Devices for," *Technologies, Innovation, and Change in Personal and Virtual Learning Environments*, p. 155, 2012.

[3] E. Ozdalga, A. Ozdalga, and N. Ahuja, "The smartphone in medicine: a review of current and potential use among physicians and students," *Journal of medical Internet research*, vol. 14, no. 5, 2012.

[4] O. I. Franko and T. F. Tirrell, "Smartphone app use among medical providers in acgme training programs," *Journal of medical systems*, vol. 36, no. 5, pp. 3135–3139, 2012.

[5] D. D. Luxton, R. A. McCann, N. E. Bush, M. C. Mishkind, and G. M. Reger, "mhealth for mental health: Integrating smartphone technology in behavioral healthcare." *Professional Psychology: Research and Practice*, vol. 42, no. 6, p. 505, 2011.

[6] M. D. Crossland, R. S Silva, and A. F. Macedo, "Smartphone, tablet computer and e-reader use by people with vision impairment," *Ophthalmic and Physiological Optics*, vol. 34, no. 5, pp. 552–557, 2014.

[7] M. Naftali and L. Findlater, "Accessibility in context: Understanding the truly mobile experience of smartphone users with motor impairments," in *Proceedings of the 16th International ACM SIGACCESS Conference on Computers & Accessibility*, ser. ASSETS '14. New York, NY, USA: ACM, 2014, pp. 209–216. [Online]. Available: http://doi.acm.org/10.1145/2661334.2661372

[8] D. Sakamoto, T. Komatsu, and T. Igarashi, "Voice augmented manipulation: Using paralinguistic information to manipulate mobile devices," in *Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services*, ser. MobileHCI '13. New York, NY, USA: ACM, 2013, pp. 69–78. [Online]. Available: http://doi.acm.org/10.1145/2493190.2493244

[9] D. Rozado, T. Moreno, J. San Agustin, F. Rodriguez, and P. Varona, "Controlling a smartphone using gaze gestures as the input mechanism," *Human–Computer Interaction*, vol. 30, no. 1, pp. 34–63, 2015.

[10] "Apple iOS Siri," https://www.apple.com/uk/ios/siri/, accessed: 2015-06-26.

[11] "Google Now," https://www.google.co.uk/landing/now/, accessed: 2015-06-26.

[12] A. Easwara Moorthy, "Voice activated personal assistant: Privacy concerns in the public space," Ph.D. dissertation, California State University, Long Beach, 2013.

[13] P. J. Treffner and R. Barrett, "Hands-free mobile phone speech while driving degrades coordination and control," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 7, no. 4, pp. 229–246, 2004.

[14] J. D. Smith and T. Graham, "Use of eye movements for video game control," in *Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology*. ACM, 2006, p. 20.

[15] R. Shaw, E. Crisman, A. Loomis, and Z. Laszewski, "The eye wink control interface: using the computer to provide the severely disabled with increased flexibility and comfort," in *Computer-Based Medical Systems, 1990., Proceedings of Third Annual IEEE Symposium on*, Jun 1990, pp. 105–111.

[16] "Tobii," http://www.tobii.com/, accessed: 2015-06-26.

[17] S. J. Luck, *An introduction to the event-related potential technique*. MIT press, 2014.

[18] "OpenBCI documentation," http://docs.openbci.com/software/03-OpenBCI_BLE, accessed: 2015-06-24.