A Google Chrome Extension for the Navigation of Active Web Pages Using Speech and Human Gaze

Renee Arianne F. Lat and John Patrick VJ. Albacea

I. Introduction

Developments in the fields of information technology and computer science have been constantly evolving in our society. We are currently experiencing vast amounts of computer systems being applied in our environment [1]. Computers are now an essential tool for communication, education, entertainment, and the like. In line with this, new types of human-computer interaction are needed. Some of these needs involve not using special additional equipment for getting different types of input [1] and to aid people who are physically disabled in using computers as well.

The most common and effective way of communicating is through speech [2]. Also, we are able to communicate more accurately using speech as the mode of communication rather than typing [3]. In our daily lives, we humans use vision and hearing as our medium of getting information about our surroundings [1]. Therefore, applying navigation through human gaze and auditory input on computer systems would provide easier access of information for all [1].

A. Significance of the Study

People who do not have any physical disabilities or do not have any difficulties in typing can easily use the keyboard and mouse combination. However, the aforementioned modality does not apply to handicapped people. There are people who suffer from an injury or are physically disabled since birth which hinder them from using keyboard and mouse combination [1].

Most people nowadays use the internet to gather information and to interact with people. Usage of the Internet has gone up to nearly forty percent(40%) [4]. People use the internet not only for leisure but also for different fields like engineering, agriculture, medicine, entertainment and the like.

In line with this, integration of both voice and gaze navigation into a general extension for web browsers that will navigate and interact on active web pages will be a huge step in the field of human-computer interaction (HCI). Web browser extensions are plug-ins that improve the interface, viewable content, or functionalities of a web browser [5]. This particular extension allow users to either use his/her voice or gaze, or both modalities (voice and gaze) to navigate and control web pages according to the users will. Using this extension will be easy since it can be installed and integrated in the browser automatically [6].

B. Statement of the Problem

One major step to the improvement of this study is making it multimodal navigation by using both human voice and gaze. Aside from that, increased responsiveness and user-friendliness will also be taken into consideration. Digital age has directed us where information gathered by mental processes can also be used for interaction [7]. People continuously strive to think of better ways to do tasks correctly and efficiently. Using voice and gaze navigation altogether, it suggests a more efficient way for people, including disabled people, to communicate and interact with our computers, particularly with web browsers which provides us the information and aid we need for our daily living and curiosities.

C. Objectives of the Study

The general objective of the study is to develop a Google Chrome extension that enables navigation on active web pages through voice and gaze navigation using an ordinary web camera and an simple external microphone. Specifically, the study aims:

1) to incorporate basic commands (normal click and scroll) using either or both modalities;
2) to incorporate advanced commands (navigation, selection, filling up of forms, etc.) using either or both modalities; and
3) to test the extension on a variety of websites.

D. Scope and Limitation

The study will concentrate on executing commands that will navigate through a page through either or both modalities (voice or gaze navigation). The extension will not be limited to navigate only one web page but it should adapt to any active web page on the browser. Additional functions were added to increase the responsiveness and user-friendliness of the extension to the user.

Concerning the voice navigation function, the extension would only be limited only to the English language and proper diction and pronunciation must be observed. As much as possible, a microphone would be needed for clearer voice input instead of using the built-in laptop microphone which is more open to unnecessary noise thus making it more prone to error.

In regards to gaze navigation, the extension is only limited to recognizing gazes without any kind of eye glasses. Also, it cannot read the gazes of people with any kind of visual impairments. For cases like this, users can always resort to using voice navigation.

The web browser extension can be applied at most to only Google Chrome version 47+.

Presented to the Faculty of the Institute of Computer Science, University of the Philippines Los Banos in partial fulfillment of the requirement for the Degree of Bachelor of Science in Computer Science.

The extension will ask the user what modality the user prefers to use (whether it will be voice, gaze, or both). Once the user chose to use the *voice* modality, all the other functionalities that require both voice and gaze will be disabled. This will also be applied if the user chose to use the *gaze* modality. Also, if the user chose to use both modalities, the functionalities for either modalities will both be enabled.

II. Review of Related Literature

A. Voice Recognition

Voice recognition is a technology that allows computers to recognize spoken words and treat it as inputs for a computer program or as text on a screen [8]. It is considered an alternative medium for typing on keyboards. Voice recognition technology can save time and energy for people especially whose jobs include encoding information [6]. Vast software programs have been developed with this technology to provide a more efficient way of putting speech into words.

Even though voice recognition technologies only became a hit in the market for the last five to ten years, the concept of voice recognition was spearheaded decades ago. Voice recognition technology started in 1952 with the system named *Audrey* developed by Bell Laboratories [11]. Given the complicated syntax of human language, developers focused on simple terms [12] thus, making *Audrey* limited to understanding numbers and a few specific people only [11]. After ten years, IBM released *Shoebox* which when compared to *Audrey*, could understand sixteen (16) English words [12]. *Shoebox*, even though it has made a huge step in the field of voice recognition, wasnt enough to cater the needs of the public [11].

In Operating Systems, voice recognition technology was also implemented by Microsoft [3]. Microsoft Word has a diction feature that lets users dictate the text instead of typing it using the keyboard. Mac OS X Mountain Lion also has a built-in diction features called *Mountain Lions* and *Nuances Dragon Dictate* [13]. In smartphones, voice recognition implementations include Googles *Voice Search* in Android and *Siri* in iOS.

In the web, *Dictation Online Speech Recognition* allows users to create e-mails and documents with the use of Google Chromes built-in speech recognition system [3]. There is an application programming interface (API) that can be supported by web browsers and that is *Web Speech API*. *Web Speech API* is developed by the W3C Speech API Community Group in JavaScript. Using *Web Speech API*, voice recognition in web browsers can be implemented and can open more possibilities for more advanced studies [14]. Examples of these implementations were from Ventocilla and Recario [3] and from Almazar and Recario [6]. They applied Web Speech API into a Google Chrome Extension as a mean to navigate through active web pages using basic voice commands (scrolling) and advanced voice commands (clicking, searching, etc). Despite of the many studies that have sprouted that implements voice recognition, Web Speech API is W3C standard which makes is more reliable to use than others.

B. Gaze Navigation

Gaze navigation is another famous technology that is more applied in games. It allows games to get sufficient information from users by tracking human gazes and using it as input [9]. With this gathered information, games can select a storyline that can suit users most and predict behavior and attention of users in games [7]. In the field of user experience, gaze navigation is also used to create more dynamic elements in a web page for the audience [10]. Software that are developed with gaze navigation technology also provides a more efficient way of encoding information into computers than the usual keyboard and mouse combination.

The major innovation in eye tracking was the creation of head-mounted eye trackers. Yarbus, a Russian psychologist, also studied eye movements by recording how people observed natural objects and scenes. As the rise of computers came, the interest for gaze tracking also persisted, making way for more interfaces and ways on how to read and track gazes with the use of computers. Most of the current and prospective aspect of eye and gaze tracking was applied in the field of games and entertainment. Studies show that action-themed games can be used as a tool in rehabilitating visually disabled or visually impaired people and improving visual attention. Also, it was confirmed that video game players enjoy more when they play with gaze tracking technology [15].

A web browser called *Weyeb* was developed to allow users use human gaze for page navigation and link selection [16]. Even though this study is helpful for people, creating a new web browser just for the sake of this function is not appealing to the market. It would be better if this kind of technology can be implemented on commonly used applications on our laptops rather than having it implemented on a new application that only has one kind of special functionality. Juruena and Abriol-Santos [7] created a Google Chrome Extension that implemented gaze navigation technology with the help of Webgazer. Webgazer is entirely written in javascript and it is the first to allow a simple web camera to detect human faces and locate eye gazes in real-time [17].

C. Multimodal Interfaces

Multimodal interfaces gives a user a choice on how to interact with a computer. Users can either choose the usual keyboard and mouse combination. But for people who are having difficulties or are physically incapable of using the usual keyboard and mouse combination, they can use other ways. Implementing multimodal interfaces on systems increases usability. The weakness of one modality can be the strength of the other [1]. It can also be used by a variety of users [1] from being physically disabled, to being visually impaired or being hearing-impaired or even users would prefer other kinds of modality rather than using the keyboard and mouse combination.

A system was developed for controlling a computer using head movements and voice commands [1]. It tracks head movements and uses voice recognition for navigation. Another application of multimodal interfaces is the development of a gesture and speech interface for controlling and interacting with three-dimensional displays and graphical objects of biomolecular systems in structural biology. With this, it simplified model the process of manipulation of biomolecular systems in structural biology. This also allows researchers to try different variations of the model without having to worry about computation and hardware limitations [18].

Despite these implementations, there are no multimodal interfaces applied in web browsers, specifically Google Chrome, implementing the two modalities mentioned above. This study will implement voice recognition and gaze navigation in a Google Chrome extension. It would not be limited to a specific web browser so that it could cater all kinds of users. This study can contribute to developing more multimodal interfaces in the web in the future.

III. Methodology

This study aims to navigate active web pages using voice recognition and gaze navigation in a Google Chrome extension. The development process, system requirements and functional requirements are discussed in this section.

A. *System Requirements*

The following tools are needed for the development of the Google Chrome extension:

1) HTML5 and CSS: HTML5 and CSS will be used for the basic layout and design of the user interface of the extension. Minimal interface, notifications and status of the extension, will be shown to the user.
2) Javascript: Javascript is the scripting language used by the web browsers. It is also used by the technology needed for voice recognition and gaze navigation. This will be the backbone in developing the extension.
3) Web Speech API: Web Speech API enables speech-input, text-to-speech output that are not available using common speech recognition software. It can both support server-based and client-based recognition and synthesis [19].
4) Webgazer.js: A real-time eye tracking Javascript library that enables common web cameras to locate eye-gazes of users on a web page. It runs only on client browser thus no data will be gathered and will be sent to the server [20].

B. *Functional Requirements*

The main functionalities of the Google Chrome extension are listed below:

Functionalities that can be applied to either voice recognition or gaze navigation

1) Settings Tab
   It can be opened by left clicking the icon that will be displayed on the interface. The following settings are as follows:
   - Opacity
   - Choose modality
   - Edit Bookmarks

   Opacity of the buttons used for gaze navigation can be adjusted to 100%, 50%, and 30% because there are users who prefer the buttons to be too obvious on the web page interface while there are other users that do not.

   Choose Modality allows the user to choose what modality to use. The user can choose between:
   - Voice
   - Gaze
   - Both

   Note that once the user chooses either voice or gaze modality, the functionalities for the other modality will be disabled unless the user selects to choose both modalities for navigation.

   Edit Bookmarks allows the user to change or delete customized keywords that the user created.
2) Scroll + *direction*
   This enables the user to scroll through the web page. The user would have to say the keyword *Scroll* together with the direction *Up, Down, Left,* or *Right* the user would want to navigate. The user can also scroll to the top or bottom of the web page by saying the keyword *Scroll to Top* or *Scroll to Bottom*. For gaze navigation, arrows on the four principal directions will be provided as a gazing point. Continuously stare at the arrows to scroll and gaze at any other part of the web page to stop scrolling.
3) Back page and Forward page
   This enables the user to go back to a web page the user has previously selected or to go forward to a web page the user has lately selected. The user need to say the keywords *Back page* or *Forward page*. For gaze navigation, the words *Back page* and *Forward page* is will be provided on the interface. The user just needs to continuously stare at those buttons to go back or forward on the browser.
4) Click + *link name*
   This enables the user to navigate another web page. For the voice recognition, the user needs to say the keyword *Click* and possible links on the web page will have numbers appearing next to it. The extension will give thirty (30) seconds to one (1) minute of waiting for what number the user will say. For gaze navigation, a *Click* button will be provided on the left side of the interface. Continuously stare at that button and wait for possible links to be highlighted. Then continuously stare at the link the user wants to open.
5) Focus + *field name*
   This functionality is applied in online forms. For the voice recognition, the user needs to say the keyword *Focus* and wait for numbers to appear on the fields detected by the extension. Then say what number user wants to fill in (e.g. Focus 1). For gaze navigation,*Focus* button on the left side of the interface will be provided.

Continuously stare at it until all the possible fields that can be filled up will be highlighted. Then stare at the desired field to be filled up.

6) Press + *button name*
This functionality allows the user to click a button on a web page. For the voice recognition, the keyword *Press* plus the label on the buttons needs to be said. If the button has more than one word, the first word will be enough for it to be pressed. For gaze navigation, continuously stare at the button until it gets clicked.

7) Open + *keyword*
This functionality allows the user to open a link in a new tab. But, this functionality is only applicable to the customized keywords that the user will save. For the voice recognition, the user needs to say the keyword *Open* and wait for a dialog box of customized keywords to appear. After that, the user can say the customized keyword of their choice. For the gaze navigation, an *Open* button on the left side of the interface will be provided. Continuously stare at it until the dialog of customized keywords appears. Then continuously stare at the desired keyword.

Functionalities that can be used by voice recognition and gaze navigation simultaneously

1) Click + *link name*
This functionality allows the user to navigate another web page. To activate this functionality, say the keyword *Click* then continuously stare on link the user wants to click.

2) Focus + *field name*
This functionality is applied in online forms. To activate this functionality, say the keyword *Focus* and continuously stare on the field the user wants to fill up.

3) Press + *button name*
This functionality allows the user to click a button on a web page. To activate this functionality, say the keyword *Press* and continuously stare on the button the user wants to press.

4) Highlight + *selection of words*
This functionality is for highlighting a selection of words on a web page. This is used for copying texts on a web page. First, the user must say the keyword *Highlight* then the user must wait for the signal from the interface before the user can say the first word where the user wants to start highlighting his/her desired text selection. Then continuously drag the users gaze to the text selection he/she wants to select. If the selection desired is already highlighted, the user can say the keyword, *Finish highlight* to end the functionality. The text will automatically be stored in a temporary clipboard and can replaced as soon as the user says the keyword *Highlight* again.

5) Save Photo
This functionality saves a photo the user wants to save from a web page. The keyword *Save Photo* must be said and wait for the signal from the interface that you can already gaze the photo you want to save. This automatically saves the photo on your computer and does not let you rename it. The user can save photos only one by one but the functionality will only stop if the user will say the keyword *End Save Photo*.

Other Functionalities

1) Save Keyword
Save Keyword lets you enter customized keyword but only for web addresses. For example, you can save facebook.com into a keyword *Facebook*. To apply this functionality, the user first need to open the link in a new tab or window. Then say the keyword *Save Keyword*. The interface will request the user to say the customized keyword for that link three (3) times. Users can only save up until ten (10) customized keywords. It can be deleted on the Settings tab. To try if its already working, the user just need to say the keyword *Open*.

C. *Measure of Effectiveness*

A modified questionnaire will be answered by testers to prove if the Google Chrome extension was effective and efficient enough for public use. This modified questionnaire is based from the Quiz for User Interface Satisfaction (QUIS) [21] and Computer System Usability Questionnaire (CSUQ) [22].

The QUIS is made up of four (4) sections evaluating the following: the terminology used and system information, learning how to use the extension, system capabilities, and the overall reaction to the extension. Analysis of data will be interpreted using Top two box (78%) scoring, which means that the score was evaluated as satisfactory if a score falls within the top two box and a score falling in the middle category is considered as a neutral response between not satisfactory (bottom two box) and satisfactory (top two box) [23]. The survey questions will be rated from 1 (being the most negative reaction to the question) to 5 (being the most positive reaction to the question). For the CSUQ, the questions are categorized into two types namely: User-friendliness, and Extension Design Efficiency. Questions 1 2,6,7,8,9,10, and 11 is applied to user-friendliness while questions 3,4,5,12, and 13 addresses extension design efficiency. Questions for both QUIS and CSUQ is shown on the Appendices.

IV. Results and Discussion

V. Conclusion and Future Work

Appendix A
Questions in Questionnaire for User Interface Satisfaction (QUIS)

A. *Terminology and System Information*

1) Use of terms throughout system
2) Terminology is easily understandable
3) Position of messages on screen
4) Prompts for adjusting input in Settings tab
5) Error messages

B. Learning

1) Overall rating for learning to navigate web pages using the extension
2) Learning to operate and to adjust the extension settings
3) Learning to navigate a web page using the extension by voice
4) Learning to navigate a web page using the extension by gaze
5) Learning to navigate a web page using the extension by both voice and gaze
6) Performing tasks is straightforward
7) Use of terms throughout system

C. System Capabilities

1) Overall extension speed
2) Extension startup speed
3) Extension response speed in recognizing voice and/or gaze
4) Failure mechanisms: Errors can occur without warnings, Error messages are displayed correctly.
5) The extension is designed for all types of users.
6) Overall usefulness of the extension for web page navigation

D. Overall reaction to the extension

1) Difficult/Easy
2) Terrible/Wonderful
3) Frustrating/Satisfying
4) Dull/Stimulating
5) Rigid/Flexible

Appendix B
QUESTIONS IN COMPUTER SYSTEM USABILITY QUESTIONNAIRE (CSUQ)

Note: The questions were rated from Strongly Disagree to Strong Agree

A. User-friendliness

- Overall, I am satisfied with how easy it is to use the extension.
- It was simple to use the extension.
- The information and warnings (such as online help, on-screen messages, etc.) provided is clear.
- It is easy to find the information I needed.
- The information provided for the system is easy to understand.
- The organization of information on the system screen is clear.
- The interface of the extension is pleasant.
- I like using the interface of the extension.

B. Extension Design Efficiency

- I can effectively navigate web pages using the extension.
- I can effectively complete my work using the extension.
- I can effectively complete my work quickly using the extension.
- The extension has all the functions and capabilities I expect it to have.
- Overall, I am satisfied with the extension.

References

[1] A Ismail, A. El Salam, A. Hajjar, and M. Hajjar, A Prototype System for Controlling a Computer by Head Movements and Voice Commands, The *International Journal of Multimedia and Its Applications*, vol. 3, no. 3, Aug 2011.

[2] M. Rafi, K. Ahmed, S. Huda, and L. Islam, Control Mouse and Computer System Using Voice Commands, *International Journal of Research in Engineering and Technology*, vol. 5, no. 3, Mar 2016.

[3] F. Ventocilla and R. Recario, Enabling Speech Navigation on Active Web Page using Google Chrome Extension, Ph.D. dissertation, 2014.

[4] Internet statistics and facts (including mobile) for 2016 hostingfacts.com, Aug 2016. [Online]. Available: https://hostingfacts.com/internet-factsstats-2016

[5] Browser extension, June 2017. [Online]. Available: $https : //en.wikipedia.org/wiki/Browser_{e}xtension$

[6] A. Almazar and R. Recario, Speech Navigation on Active Web Page using Google Chrome Extension and Web Speech API, Ph.D. dissertation, 2016.

[7] C. Juruena and K. Abriol-Santos, eyeGalaw: A Google Chrome Extension for Webpage Navigation through Human Gaze, Institute of Computer Science, UPLB. Special Problem, 2017, unpublished paper.

[8] S. Halim and W. Budiharto, The Framework of Navigation and Voice Recognition System of Robot Guidance for Supermarket, *International Journal of Software Engineering and Its Applications*, vol. 8, no. 10, pp. 143-152, 2014.

[9] P. Isokoski, M. Joos, O. Spakov, B. Martin, Gaze Controlled Games, *Universal Access in the Information Society*, vol. 8, no. 4, p. 323337, May 2009.

[10] Design, User Experience, and Usability: Interactive Experience Design. Springer, 2015. [Online]. Available: https://books.google.com.ph/books?id=e0UqBAAA QBAJ

[11] A Brief History of Voice Control, Oct 2016. [Online]. Available: https://medium.com/@joshdotai/its-all-in-the-voicefba7e5a032fa

[12] Speech Recognition Through the Decades: How We Ended Up With Siri, Nov 2011. [Online]. Available: $http : //www.pcworld.com/article/243060/speech_{r}ecognition_{t}hrough_{t}he_{d}ecades_{h}$

[13] Mountain Lion Dictation versus Dragon Dictate, Sept 2013. [Online]. Available: https://macworld.com/article/2014417/mountainlion-dictation-versus-dragon-dictate.html

[14] Using Voice to Drive the Web: Intorduction to the Web Speech API, Sept 2013. [Online]. Available: http://www.adobe.com/devnet/html5/articles/voiceto-drive-the-web-introduction-to-speech-api.html

[15] A. Mohamed, M. da Silva, and V. Courboulay, A History of Eye Gaze Tracking, *Rapport Interne*, 2007.

[16] M. Porta and A. Ravelli, Weyeb, an eyecontrolled web browser for hands-free navigation, *2009 2nd Conference on Human System Interactions*, 2009.

[17] A. Papoutsaki, P. Sangkloy, J. Laskey, N. Daskalova, J. Huang, and J. Hays, Webgazer: Scalable Webcam Eye Tracking Using User Interactions, *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, pp. 3839-3845, 2016.

[18] R. Sharma, V. Pavlovic, and T. Huang, Toward Multimodal Human-Computer Interface, *Proceedings of the IEEE*, vol. 86, no. 5, May 1998.

[19] Web Speech API Specification, Oct 2012. [Online]. Available: https://dvcs.w3.org/hg/speechapi/raw-file/tip/speechapi.html

[20] WebGazer.js Democratizing Webcam Eye Tracking on the Browser, 2016. [Online]. Available: https://webgazer.cs.brown.edu

[21] J. R. Lewis, Ibm computer usability satisfaction question- naires: Psychometric evaluation and instructions for use, *International Journal of Human-Computer Interaction, 7:1, 57-78.,* 1995.

[22] J. P. Chin, V. A. Diehl, and K. L. Norman, *ACM*, 1988.

[23] "How to Interpret Survey Responses," 2011. [Online]. Available: https://measuringu.com/interpret-responses/