Voice-Activated Mouse Controller

Mallika Rai
Stony Brook University, New York

December 16, 2021

1 Abstract

The last few decades have witnessed an increased dependence on software in education, employment, e-commerce and entertainment. However, this increased dependence poses the challenge of inaccessibility for a significant population with visual and motor disabilities. Motivated by the need for more accessible, affordable and inclusive software, in this have come up with a Voice-Activated Mouse Controller (VAMC) to provide people with such disabilities the means of controlling mouse navigation and clicks on their system using voice. Using the voice-activated mouse, users will be able to perform basic operations such moving the cursor, right-click, left-click, double-click and open various applications on their systems.

2 Introduction

Human-computer interaction is a discipline that focuses on technology to facilitate seamless interaction between humans and a computer to perform various tasks. Being able to control your mouse with just your voice is an example of one such task. Technologies such as speech recognition have made a lot of headway over the past couple of years and come to the rescue of people with various physical, cognitive and motor disabilities in facilitating interaction. Using speech-recognition based technologies, with simple voice commands, people can control technology around them. This forms the basis for a voice-activated mouse. The voice-activated mouse will be able to ensure greater participation and inclusion of people with disabilities in our society by enabling interaction between them and their computer systems.
The voice-controlled mouse presents several benefits for the abled and disabled alike:

- Enables people with disabilities- By allowing users to control their mouse hands-free using voice commands, the voice-activated mouse provides an effective means of interaction for users with various motor, cognitive and visual disabilities, thus encouraging their participation and inclusion in the society.
- Improved productivity- VAMC allows users to control trivial tasks requiring mouse actions hands-free, thus allowing them to multitask and focus on more important tasks simultaneously.
- Eradicates language barriers- VAMC can be programmed to take speech input in the user's choice of language, thus enabling even users with a language handicap to interact effectively with their system.
- Personalized user experience- VAMC can

be tailored to perform and repeat tasks specific to each user, therefore serving as a virtual assistant with a personal touch

3 Related Work

Human-computer interfaces facilitate communication, assist on the exchange of information, process commands and controls and perform several additional interactions. Spoken natural language is more user-friendly mean of interacting with a computer. From the human perspective point, this kind of interaction is easier since it does not urge humans to learn additional interactions. Humans can rely on natural ways of communications instead. Human-computer interaction varies from understanding simple commands to extracting all the information in the speech signal such as words, meanings and emotions of the user. To develop an interface with natural language understanding ability, several factors arise and must be taken into account such as dealing with the ungrammatical nature of many spoken utterances, the detection of problems in speech recognition, and the design of intelligent clarification dialogues.

There are also verbal interaction problems such as background noise, word echoing and repetition, and different and background sound sources overlapping the speech. Solution for these problems must be developed. Most state-of-the-art of the human-computer verbal interaction based systems relies on using simple voice commands, hence may not be considered as complete speech recognition systems. Having these concepts in mind and addressing human-computer verbal issues, the speech products should be well developed and be resilient enough for effective human-computer verbal interaction. The system we are proposing here consists of providing an augmented speech dialogue structure used as and tested to study verbal human-computer interaction.

4 Implementation

This section summarizes the technologies and algorithm used to develop the VAMC and its features.

4.1 Technologies

The voice-activated mouse was primarily developed using two python libraries:

1. SpeechRecognition: This python module works by converting voice inputs from physical sound (from recorded audio or microphone) to an electrical signal with a microphone, and then to digital data with an analog-to-digital converter. Once digitized, several models can be used to transcribe the audio to text.

2. PyAutoGUI: This library lets python scripts control mouse and keyboard to automate interactions with other applications. It offers support for Windows, macOS, and Linux operating systems has several actions such as moving the mouse and clicking in the windows of other applications, sending keystrokes to applications (for example, to fill out forms), take screenshots, and given an image (for example, of a button or checkbox), and find it on the screen, locate an application's window, and move, resize, maximize, minimize, or close it (Windows-only, currently) and display alert and message boxes.

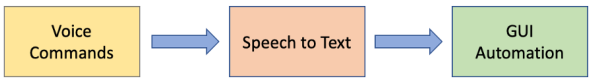

Figure 1: *Voice-activated mouse flow diagram*

4.2 Features

Using VAMC, a user can control the mouse pointer hands-free by providing voice commands. These voice commands are validated and mapped to their corresponding gestures

automated by VAMC, as listed in Figure 2.

COMMAND	SUB-COMMAND	ACTION
Move Up		Move cursor up
	Faster	Move cursor up faster
	Slower	Move cursor up Slower
	Stop	Stop moving cursor up
Move Down		Move cursor down
	Faster	Move cursor down faster
	Slower	Move cursor down Slower
	Stop	Stop moving cursor down
Move Right		Move cursor right
	Faster	Move cursor right faster
	Slower	Move cursor right Slower
	Stop	Stop moving cursor right
Move Left		Move cursor left
	Faster	Move cursor left faster
	Slower	Move cursor left Slower
	Stop	Stop moving cursor left
Open Chrome		Open Chrome application
Open Spotify		Opens Spotify application
Play/Stop Playing		Play/Pause music/video
Click/ Left click		Left click once
Double click		Left click twice
Right click		Right click once

Figure 2: *Voice commands supported by VAMC*

4.3 Algorithm

Below outlined is the algorithm behind automating the mouse pointer.

1. Voice input: Using the microphone, user provides a voice command to activate the mouse pointer that is captured by the program. This is achieved using Python's pyaudio module.
2. Convert input speech to text: Once the voice input is captured, using google online API supported by Python's Speech Recognition module, the audio input is converted to text.
3. Search in Commands: The converted text command is mapped to exiting commands that VAMC is programmed to accept.
   a. if command is found, perform the corresponding GUI action
   b. if command is not found, the user is prompted again for input
4. The above steps are repeated until the user issues a STOP command

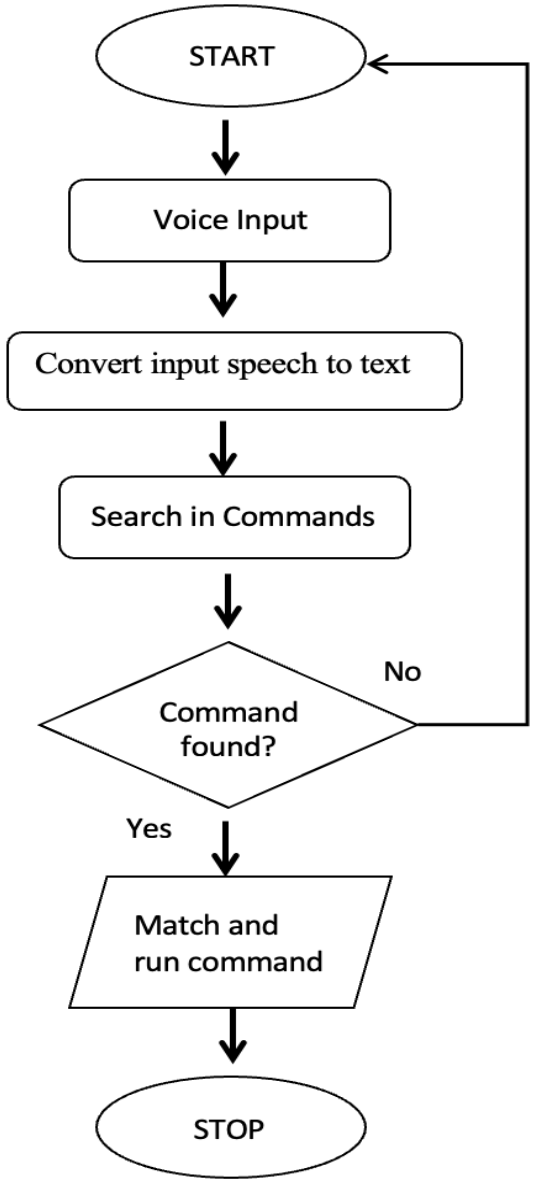

Figure 3: *VAMC algorithm flow*

5 Validation

In order to test and validate VAMC application, Heuristic Evaluation was performed with a group of 3 users. Each of the users evaluated and confirmed that the product conformed to the following heuristics:

1. **Visibility of system status**: Users confirmed that for each input, they could see the system status clearly, that is, if the input was captured or not.
2. **User control and freedom**: Users felt that the system was in their control as

the system would suspend when no input was provided and would terminate when the user would give command STOP.

3. **Error prevention**: The system prevented erroneous commands and results by relaying back to the user in case an unregistered command was provided.
4. **Recognition rather than recall**: The application provides users a list of commands to choose from instead of them having to remember the commands beforehand
5. **Flexibility and efficiency of use**: The users had the flexibility to alter the input commands they wished to configure to run one or more commands. The user tested the system by modifying "Open Chrome" command to "Chrome" command to perform the same action.
6. **Help users recognize, diagnose and recover from errors**: Users were able to efficiently identify the error based on the error message that appeared on the screen and repeat their input

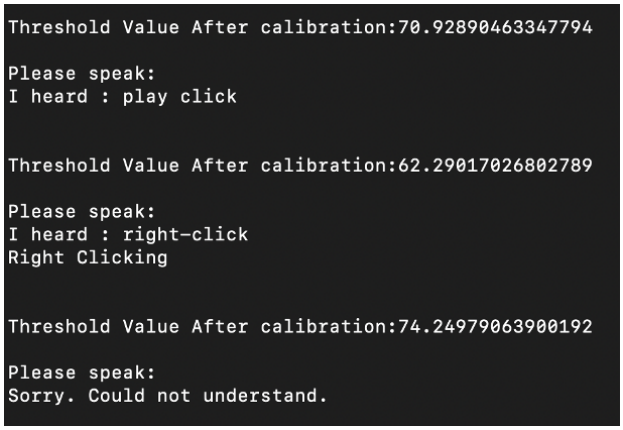

Figure 4: *VAMC Heuristic Evaluation - System status and recovery from error*

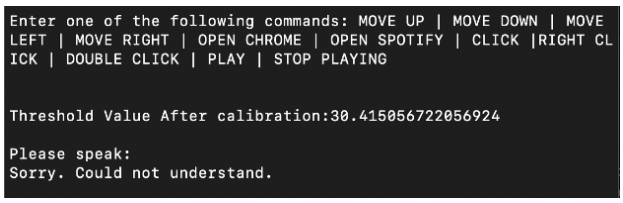

Figure 5: *VAMC Heuristic Evaluation- Recognition over recall*

6 Future Work

In future, in addition to adding more commands to open/close other applications, the project can be extended to include the following features:

1. Platform independent - Currently the voice-activated mouse has been developed to perform various actions and gestures across MacOS. However, Python's speech recognition and PyAutoGUI libraries provide support for both Mac OS and Windows and therefore, the voice activated mouse can be extended to a Windows system as well
2. Language agnostic – VAMC can be programmed to take user's language as input or issue commands in user's language of choice, thus making it feasible for anyone with a language barrier to use just as well.
3. Voice search - Currently, VAMC is limited to performing navigation and clicks across the system and opening applications. VAMC can be extended into a complete voice-assistant for your system by incorporating commands to perform search within a browser using Speech Recognition technology or set various reminders.

7 Conclusion

We were successfully able to implement a voice-activated mouse using Python's SpeechRecogniton and PyAutoGUI modules. Using the voice-activated mouse, users were able to control various gestures on their systems such as moving the mouse pointer up, down, left or right, clicking, opening applications on the system, etc. Because of its ease of use and low setup cost, the voice-activated mouse can find a wide range of applications in the fields of education, healthcare, banking and more.

8 Acknowledgements

I would like to express my sincere gratitude towards Prof. Xiaojun Bi who gave me the opportunity to work on this project. Under his guidance, I gained in-depth understanding of various concepts taught in the course CSE518 Foundations of Human Computer Interaction and was able to successfully apply them through this project.

I would also like to acknowlege my TAs for the course, Shahreen Salim, Dan Zhang and Yoonsang Kim, for their constant inputs and feedback on how to improve the project. These inputs were vital in making the project what it is.

Last but not the least, I would like to acknowledge my peers for taking the time to test and evaluate my project. They provided some very insightful feedback without which I would not have been able to make the user experience of the Voice-Activated Mouse Controller as seamless and interactive.

Overall, through this project, I learnt the importance of constant and iterative user feedback in the life cycle of a project involving interaction between humans and computers.

9 References

1. Speech Recognition 3.8.1 manual. https://pypi.org/project/SpeechRecognition/
2. PyAutoGUI Documentation. https://pyautogui.readthedocs.io/en/latest/
3. Sarita and Kiran Kumar Kaki. "Mouse Cursor's Movements using Voice Controlled Mouse Pointer". https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.402.6692&rep=rep1&type=pdf
4. Kevin Christian, Bill Kules, Ben Shneiderman and Adel Youssef. "A Comparison of Voice Controlled and Mouse Controlled Web Browsing". https://www.cs.umd.edu/ ben/papers/Christian2000comparison.pdf
5. Mahammad Rafi, Khan Sohail Ahmed, Shaikh Huda, Lutful Islam. CONTROL "MOUSE AND COMPUTER SYSTEM USING VOICE COMMANDS". https://ijret.org/volumes/2016v05/i03/IJRET20160503077.pdf
6. Thatcher, Jim; Jenkins, Phil; and Laws, Cathy. "IBM Special Needs Self Voicing Browser." W3C Workshop on Voice Browsers, Cambridge, MA (1998) http://www.w3.org/Voice/1998/Workshop/PhilJenkins.html
7. Robin, Michael and Jim Larson. "Voice Browsers: An introduction and glossary for the requirement drafts."W3C Working Draft, (1999) http://www.w3.org/TR/voice-intro.
8. Frank Loewenich and Frederic Maire, (2008) "Motion-Tracking and Speech Recognition for Hands-Free Mouse-Pointer Manipulation", Queensland University of Technology Australia, ISBN: 978-953-7619-29-9, pp. 550.