Kaleidoscope

Volume 1 Article 10

2002

Finger Mouse and Text-To-Speech Application as Additions to the Smart Wheelchair

Karla Conn
University of Kentucky

AUTHOR

Karla Conn

NSF Summer Undergraduate Fellowship at the University of Pennsylvania

I am currently a senior in the Department of Electrical and Computer Engineering. In 2001, the University of Pennsylvania accepted me into the SUNFEST (Summer Undergraduate Fellowship in Sensor Technologies) program — a 10-week REU (Research Experience for Undergraduates) program. During the summer, I completed the research described in my paper. The REU program at UPenn, inspired me to think about research as a future career. I plan to finish my B.S. degree in Electrical Engineering in December 2002 and enroll in graduate school for the following fall semester. I have not chosen a specific university, but I would like to find an Electrical Engineering or Biomedical Engineering program focusing on robotics or prosthetics. I am currently working in a biomedical lab at UK with Dr. Ranu Jung. I have been an active member of SWE (the Society of Women Engineers) for four years. In addition, I held the position of chair of the UK student chapter of the IEEE (Institute of Electrical and Electronics Engineers) for the past year. Through IEEE, I presented my research work in a technical paper competition for undergraduate research projects at the 2002 IEEE Southeastern Conference. I won second place out of sixteen entries from universities such as Duke, South Carolina, Virginia Tech, and Georgia Tech. As a non-engineering outlet, I am extremely involved in Alpha Phi Omega, a national co-ed service fraternity. I am now looking forward to graduate school, earning my Ph.D., and becoming a university professor.

Faculty Mentor: Dr. Ranu Jung, Associate Professor of Biomedical/Electrical and Computer Engineering

There is a tremendous need for providing improvement in assisted living to people restricted in their freedom of movement and impaired in communication abilities. In her rehabilitation engineering REU project, conducted under the guidance of Dr. Jim Ostrowski at the University of Pennsylvania in the summer of 2001, Karla developed and implemented a text-to-speech hardware/software attachment for wheelchairs that targets this need. The product is both useful and innovative. Not only can people with speech impairment use it, but also, because it is a system mounted on a wheelchair, it provides “freedom of speech and mobility.” Since the completion of this work, Karla has independently written a clear and thorough report describing the development, implementation, and use of the system.

Finger Mouse and Text-To-Speech Application as Additions to the Smart Wheelchair

ABSTRACT

The smart wheelchair project is a unique investigation into the possibilities of helping the impaired navigate in a mobile chair. Many disabled people who need the help of a wheelchair to move about also need help communicating orally. This project allows the “walking” wheelchair to do some “talking.” A communication program for the wheelchair was developed, as well as a finger mouse that was implemented into all the programs. The finger mouse is a switch button small enough to wear on one’s finger. When the button is pressed, a signal is sent out from the transmitter and picked up by the receiver, which sends a signal through the parallel port of the computer to execute the desired application. The communication program is a speech program that speaks text messages. The finger mouse and speech application are connected through a communication display interface — a page of icons. When the mouse is clicked over an icon linked to a particular phrase, the mouse, the display, and the speech software work together to speak the phrase. Thus, the communication program gives the freedom of speech to anyone using the wheelchair for free range of motion.

INTRODUCTION

This work was completed during the summer of 2001 at the University of Pennsylvania as part of an REU (research experience for undergraduates) program. Dr. Jim Ostrowski, associate professor in the Department of Mechanical Engineering and Applied Mechanics supervised the project as the faculty mentor. Dr. Ostrowski, overseer of many projects in the GRASP (General Robotics, Automation, Sensing, and Perception) lab, along with a group of Penn graduate students, developed the "smart" wheelchair project during the eight months prior to this work.

A text-to-speech application was designed and implemented as a supplementary application for the mobile wheelchair. A finger mouse was completed that signals to the computer the function the user wants to execute. The speech program uses the computer's default voice to speak certain phrases when the finger mouse selects the appropriate icon on the communication display. The finger mouse can also be integrated into the navigation program for the wheelchair.

BACKGROUND

The smart wheelchair project entails the development of an autonomous wheelchair that interweaves human and computer control for navigation and communication. The wheelchair is equipped with a camera and projector mounted above the chair and a desk surface placed across the arms of the chair. The projector displays the picture of the computer program onto the desk surface so that the camera can relay to the computer where it "sees" the person's hands on the desk surface (see figure 1)(Rao, 2001).

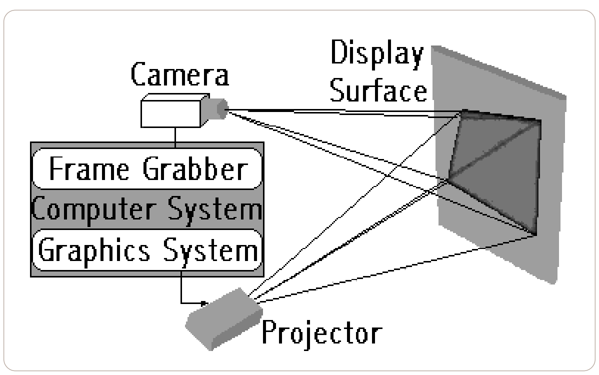


Figure 1. Interactive system for the wheelchair. The camera, the computer program, and the projector work together to display the image of the computer program onto the surface.

Because the camera is sensitive to light and dark images, the mobile chair is operated by having a person cover white squares on a display board with his or her black-gloved hands (see figure 2)(Rao, 2001).

When a black glove covers a white square, the camera and computer can interpret which square is covered. The program is coded to execute an application when more than half of an "application square" is covered along with more than half of the "click" square (comparable to clicking a computer mouse when the cursor is covering the program icon to be executed). The operator first covers the square signifying the function he or she wants to execute and then covers the click square to execute that function. Covering two squares is not the most efficient way to select an operation. Therefore, the first part of the project involved developing a "mouse" small enough to wear on the user's finger, to eliminate the click square and black gloves. The finger mouse is a switch button that signals the computer when it is pressed; it was tested with the speech program.

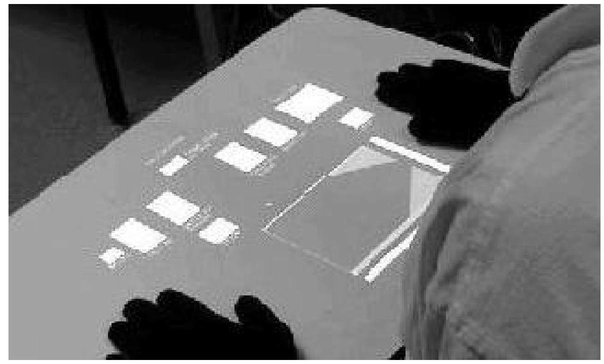


Figure 2.
The navigation display for the mobile wheelchair is shown with a person's black-gloved hands. The black-gloved hands cover white squares on the desk surface to control the computer programs.

OVERVIEW

Using both hands to execute an application is not ideal. Therefore, a small finger mouse is very useful to the entire wheelchair project, because it minimizes the space required to interact between the user and the computer. This mouse takes on the role of the previously used "click" square and the black gloves but uses less space. Instead of using two hands, a person now only needs to use his or her finger. The first mouse was very susceptible to noise from the machinery in the lab and did not work properly. This project redesigned the wireless mouse to make the signal more resilient.

A new program for the wheelchair was also developed that uses functions in the Microsoft Speech software design kit (SDK) 5.0 to "talk." Many disabled people who need the help of a wheelchair to move about also need help communicating orally. This project adds a second feature to the chair and a new display for interaction, similar to the navigation display (see figure 2) (Rao, 2001). Icons on the new communication display are linked to phrases. The computer code and the SDK allow these phrases to be spoken when the mouse is clicked over the associated icons. This project will open the gateways of communication at the touch of a button.

FINGER MOUSE

For the finger mouse portion of the project, the preliminary finger mouse — designed by Terry Kientz, a lab technician in the MEAM (Mechanical Engineering and Applied Mechanics) Department at the University of Pennsylvania — was studied for guidance toward the next generation.

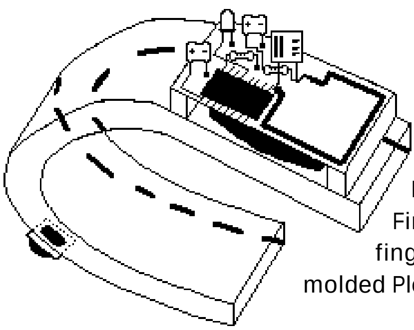


Figure 3. First version of the finger mouse with molded Plexiglas design.

The preliminary version of the finger mouse was crafted out of Plexiglas (see the photo on page 32). A finger fits inside the curved glass and rests on a push button underneath that triggers the transmitted signal. The battery and circuit board with the transmitter sit atop in this design. This first version had little room for adjusting the fit. The new mouse (see figure 4) has a form-fitting design made of brass, Velcro, and Plexiglas. A finger fits inside the curved brass, and the Velcro strap is fitted around the finger to secure the finger comfortably in place. Therefore, the finger mouse adjusts to any size finger.

In addition, the switch button is connected to a flexible piece of brass that curves around the tip of the user's finger to adjust to the most comfortable feel. The push button is positioned underneath the brass so that the pad of the user's finger pushes on the button. The battery and the small circuit board with the transmitter are mounted on Plexiglas and incased in a black box as the top of the finger mouse design. This black box interacts with the camera in the same way the black gloves did previously, but

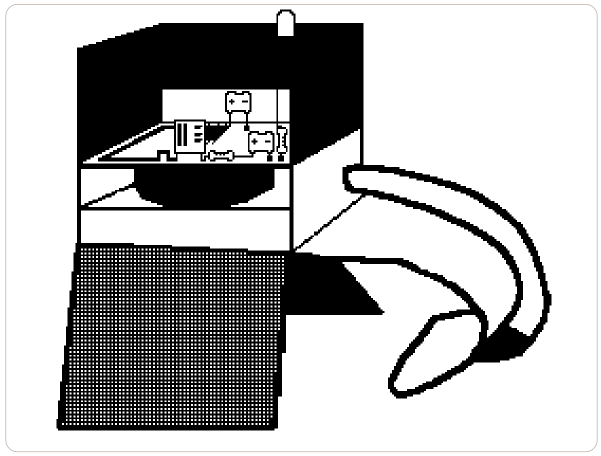


Figure 4. Current version of the finger mouse with brass and Velcro design.

now the area is condensed into a 1" by 2" rectangle. A red LED (Light Emitting Diode) pokes out of the top of the black box and flashes when the button is pressed as a feedback indicator for the user.

When the switch button is pressed, a signal is sent out from the transmitter and picked up by the receiver. A Programmable Intelligent Computer (PIC) microprocessor chip and the transmitter in the circuit are programmed to transmit a unique signal from the finger mouse. The receiver, with the help of another PIC chip, recognizes this signal. When the receiver detects the correct transmission, one bit of data is sent through the parallel port of the computer to execute the desired application. The program code reads this bit, and the application executes. The finger mouse can be used in all the wheelchair applications, including the communication program.

There are several advantages to the size and independence of the wireless finger mouse. First, a person sitting in the wheelchair needs to use only one hand when selecting an icon. This single-handed interaction frees up the other hand for making gestures, such as waving, while the computer says "hello" to a friend. The finger mouse is a compact circuit and takes up minimal space on the desk surface, providing area in the limited space of the desk surface for more icons. In addition, because the finger mouse fits on the end of one hand's index finger with a switch button placed under the pad of the finger it contributes to *true* "point-and-click" actions. When the person wants the computer to speak, he or she needs only to *point* with a finger and *click* the button.

When the switch button is pressed, a signal is sent out from the transmitter within the mouse circuit board. This transmitter is connected to a microprocessor PIC chip, which creates a unique signal from the transmission. A microprocessor is in effect a very simple computer. The PIC chip is programmable and, therefore, adaptable to numerous engineering applications, including computers. Memory addresses in the chip, which can be written to or read from, are used for storage. In addition, the 1/3" square chip has input and output pins that feed data in and extract data out after the bits have been manipulated by program code (Microchip, 1998).

Every microprocessor has its own programming language, called an assembly language, though all are similar. Becoming fluent in the chip's particular language is essential for a given project and the first step toward a finished project. The programmer must first fully understand the language commands, and then he or she can adapt the capabilities of the chip to fit his or her needs.

Criteria for the mouse design involved creating a unique signal that could be recognized by the receiver. A square waveform signal with a 50% duty cycle and 2 kHz frequency was chosen to release four pulses, signaling high (2.5v) for one fourth of a millisecond and then low (0v) for one fourth of a millisecond and then repeated three more times (see figure 5). A larger

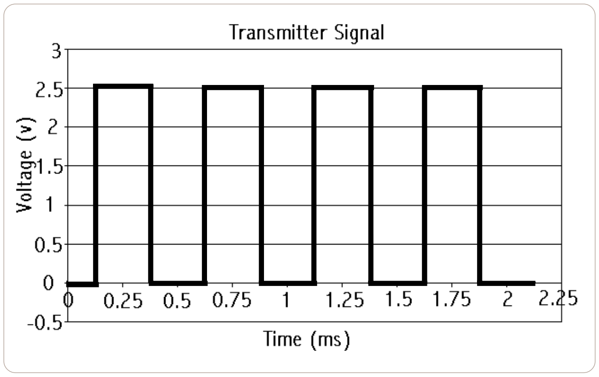


Figure 5. Graph of Transmitter Signal.

number of pulses could have been used, as well as a longer or shorter pulse length, but a manageable frequency and number of pulses were chosen for troubleshooting and preliminary design purposes. If desirable in the future, perhaps to avoid interference, the PIC code can change the parameters of the signal.

In addition to sending a unique signal, the transmitter continuously repeats the signal while the button on the mouse is held down, in order to prevent missing a click. If the string of four pulses were sent out only once when the mouse was clicked, the receiver might not recognize the single transmission correctly, thus missing the click and not executing the desired application. Therefore, the signal is transmitted continuously to ensure that the signal is received each time the mouse button is clicked. The transmitter and PIC chip work together to produce a neat, clean, and stable signal.

Once the PIC chip sends its unique signal through the transmitter, the receiver must identify it. Any signal that is not the finger mouse signal must be disregarded. Therefore, another PIC chip in the receiver circuit is used to recognize this unique finger mouse signal.

The length of the signal's high (2.5v) time and low (0v) time is measured, and the number of pulses is counted.

The measured length of the pulses is compared to high and low limiting bounds, a little above and below one fourth of a millisecond. Any pulse of smaller or larger width is disregarded as noise. After a pulse clears the length tests, the PIC chip begins to count the number of pulses of correct length. Any signal without four pulses of approximately one half of a millisecond each is ignored. Therefore, the receiver ensures that the unique finger mouse signal is properly detected.

When the counter in the PIC chip tallies four accurate pulses, the receiver has interpreted the correct signal from the transmitter, and the counter is reset to zero to await the next set of pulses. At the same time, the receiver sends a one (high) bit of data out to the parallel port on the computer (Walker, 2001). This bit of data is represented by a variable called "Mouse" in the computer program code. Mouse is originally set to zero, and the code is written to look for changes from zero (low) to one (high). The mouse click signal is the first of the two actions that must be performed before the computer speaks. The second half involves covering the squares on the communication display.

COMMUNICATION DISPLAY

The second phase of the project entailed developing the communication board to be displayed on the desk surface. The board has a page of square bitmap images with labels such as “Hello,” “How are you?,” and “What is your name?” for the finger mouse to select (see figure 6) (Barry 2001). Microsoft Virtual C++ code generates this page with commands from OpenGL to import the images. Currently twelve images can be displayed without cramping the space; this allows icons for several phrases and a few program functions such as “Exit Program.” A future goal for the communication board is to complete the code to “flip” to additional pages, but the current page suffices for the mouse and speech applications.

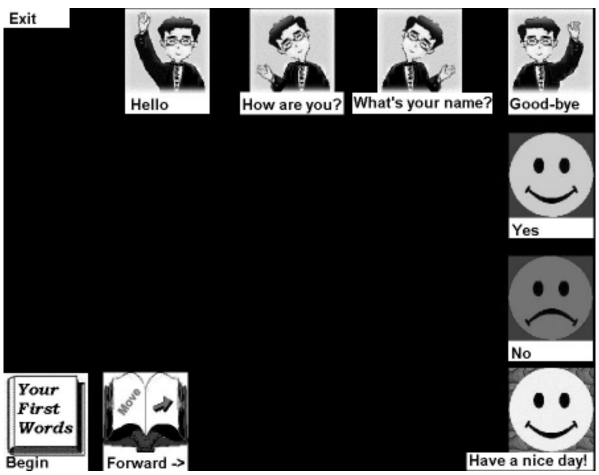


Figure 6. Introduction page of communication display.

Using OpenGL commands, the communication display imports bitmap images onto the rectangles in the display. OpenGL includes a process called texture mapping that pastes images onto specially defined rectangles. Texture mapping allows a user to map a picture to any type of object created in OpenGL. As long as the object is well defined, the picture can be wrapped around it without having to define the picture any further. For example, commands in OpenGL allow a programmer to paste a 360-degree picture of a room onto a goblet created by OpenGL commands. Because the goblet is defined by the OpenGL code, the picture can be pasted so that the goblet perfectly reflects the room as if it were placed on a table in the middle of it. The requirements for the wheelchair displays did not call for such elaborate images. Pasting a flat, square bitmap image onto a flat, square texture rectangle sufficed (Woo, et al., 1997). The code for loading the bitmaps onto the texture rectangles is included in the Web version of this article.

The seven text phrases on the display are linked to code that speaks the words when the user selects them; and the three function rectangles execute individual commands. If the “Exit” rectangle is covered when the finger mouse is clicked, the program closes. The two icons in the bottom left corner are for an extended vocabulary. They compare to the “Back” and “Forward” buttons on a Web browser. Selecting them will flip through the next display pages of icons and phrases.

The code for flipping pages is only in the first stages of development and is not currently implemented in the speech code and display. However, how the extra pages will work with the first page of dialogue has been considered, and a second page of dialogue is finished (see figure 7) (Smiley 2001). This second page consists of phrases a user might say when navigating the wheelchair. Determining how to run the speech and navigation programs simultaneously are future goals of the wheelchair project, but first the two pages of the speech display will be linked.

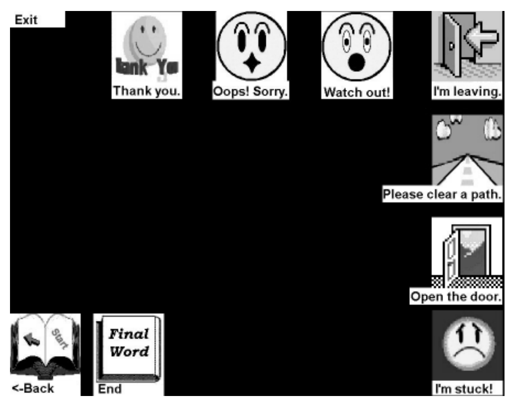


Figure 7. Navigation page of communication display.

On the first display, the “Begin” rectangle is not linked to any function; it is just a placeholder to indicate that the user is on the first page of phrases (see figure 6) (Barry 2001). The “Begin” rectangle is, therefore, like the grayed-out “Back” button on a browser window. The “Forward->” rectangle is linked to the next page of phrases, and its icon displays the topic of those phrases. The second page has an active “<-Back” rectangle with the appropriate icon displaying the topic of the previous page. The last page of phrases has an active “<-Back” rectangle and an inactive “End” rectangle (see figure 7) (Smiley 2001). This

"End" rectangle indicates the end of the pages, just as a grayed-out "Forward" button on a Web browser tells the user that there are no following pages to view.

TEXT-TO-SPEECH APPLICATION

The final word on the project comes from a voice synthesizer in the computer. Using the tools in Microsoft Speech SDK 5.0, the computer program commands the computer to speak the phrases on the communication board. The SDK provides a sample text-to-speech (TTS) application — TTSApp — that speaks words and phrases written in text files. Using the TTSApp as an example, the small white rectangles on the display link with text files to develop the "talking" part of the wheelchair. The user clicks the finger mouse to select the desired icon. Then the phrase associated with that icon is "spoken."

Microsoft's software design kit (SDK) works with the computer's voice in a manner similar to how OpenGL works with bitmap images. The kit provides a library of functions that can be implemented in C++ programs for speech applications. There are several possibilities to choose from the SDK. One could develop a speech recognition program that interprets what a user is speaking into a computer's microphone, but this software is limited by the different dialects of users in every city, country, and continent. Another, more reliable application of the kit is the text-to-speech application, used with the communication displays. These commands allow a user to create a program that speaks any text message. The core code for executing a text message is included in the Web version of this article.

CONCLUSION

The finger mouse, communication display, and speech program tested successfully and functioned fully. The new mouse design is not affected by any of the first generation's noise interference problems. The finger mouse consists of a compact design with a clear signal and fits our needs perfectly. The communication display is completely interactive and adjustable. Expansions can be made for flipping between pages of phrases, but the two individual pages are an excellent freedom of speech to anyone using the wheelchair for free range of motion.

ACKNOWLEDGMENTS

Thanks to Dr. Jim Ostrowski for being a very supportive and encouraging advisor as each part of this project was completed. Terry Kientz was the essential key for finishing the second generation of the finger mouse. Many thanks to all the great students and faculty in the GRASP lab for their fellowship, direction, and expertise – Rahul Rao, Bill Sacks, Sarangi Patel, Ray McKendall, Dan Walker, and C. J. Taylor. The National Science Foundation and Microsoft supported this work.

REFERENCES

Rao, R. S. (2001). "The Smart Wheelchair — An Overview." http://www.cis.upenn.edu/smartchair/.

ABACOM Technologies, *AM Transmitter/Receiver Module Manual,* Etobicoke, ON, Canada.

Microchip Technology, Inc., *PIC16F8X Data Sheet and Instruction Set Summary,* 1998.

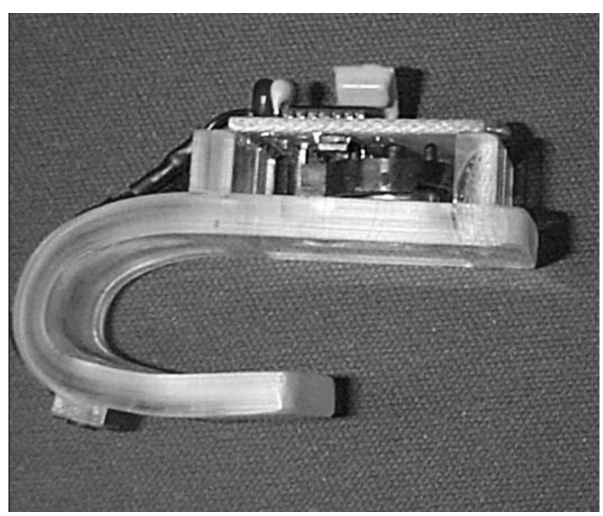
Walker, D., "Remote Manipulation of Mobile Robots Using Stock Radio Components," March, 2001.

Woo, M., Neider, J., and Davis, T. (1997), *The Official Guide to Learning OpenGL, Version 1.1,* Addison-Wesley, Reading, Massachusetts, 2nd ed.

Barry's clipart server. (2001) http://www.barrysclipart.com/

Smiley face clipart. (2001) http://www.busprod.com/nlaird/smileycol.htm

Microsoft Speech SDK 5.0 Help document, sapi.xsd.



Preliminary finger mouse