A Tongue Based Input Device

Daniel Kelly
University OF Washington
Electrical Engineering
AE100R Paul Allen Center
Seattle WA 98195 USA
kellyd22@u.washington.edu

Julia Schwarz
University OF Washington
Computer Science
CSE505 Paul Allen Center
Seattle WA 98195 USA
julenka@u.washington.edu

Copyright is held by the author/owner(s).

Abstract
In this paper we describe and evaluate a device that enables users to provide computer input using their tongue. Unlike many existing devices, our retainer-based device is easy to remove, and is accurate enough to enable cursor movement and text input.

Keywords
Input technique, assistive technology

ACM Classification Keywords
H5.m. Information interfaces and presentation (e.g., HCI

Introduction
Much of modern life is completely dependent upon the use of one's hands. Common examples are a keyboard and mouse for computer input, the keypad on a cell phone, and the remote control for a television. For severely handicapped individuals, the use of these electronic devices for either entertainment purposes or job related tasks is all but impossible. Technologies designed around the unique capabilities of handicapped people greatly improve the individuality of each person by enabling him or her to perform more tasks without assistance. Such technologies enable an increase in self-supportiveness and the possibility of employment.

There are many versions of these assistive technologies intended for controlling electronics, but few have found universal use. The most successful examples are the chin control system, in which a user provides input by pushing his or her chin against a joystick, and the sip-n-puff switch, where input signals come from sipping or blowing into a tube. Both of these systems have commercial implementations.

There are also several movement recognition systems. One implementation uses a camera to track and recognize head and facial movements[10]. Gaze tracking attempts to infer what a user is concentrating on by recognizing what he or she is looking at[1,6,8,12]. Movements of the eyes could also be used as an input technique is this scenario. All of these ideas suffer from the fact that it is difficult for the system to detect, or burdensome for the user to signal, when he or she is using the system or when he or she is simply moving in a natural manner. Also, natural movements during input times could disrupt the signal analysis.

Brain-computer interfaces are another technique for providing assistive technologies[2,7]. These devices attempt to monitor what a user is thinking about by detecting electrical neuron signals in the user's brain. These devices either rest on top of, or inside, the skull. They can potentially provide assistive technologies to a wider range of disabled people, but are currently limited by the large amount of signal processing necessary and thus slow response times.

Few of these devices have found wide success outside of the lab. The wheelchair is the quintessential assistive technology for movement around a physical space. The wheelchair of the electronics space has yet to be discovered.

The Tongue as an input device

The device proposed in this paper uses the tongue as an input technique. This idea has merit because the majority of severely disabled people retain use of their tongue. Dexterity, nimbleness, and sensation are also factors to consider. An analysis on how different organs map to the sensory and motor cortexes of the brain shows that the lips are the most sensitive organ and the hands the most dexterous. The tongue has sensitivity similar to the hands and dexterity similar to the feet.

Other researchers have also used the tongue as an input technique. Huo, Wang, and Ghovanloo attached a magnet to the tongue and then used Hall Effect sensors to monitor how magnet fields changed as a user moved his or her tongue around[3]. This technique proved effective, but still suffers from the fact that a user must attach a magnet to his or her tongue. Qiyu and Budinger setup a wireless joystick that could be maneuvered with the tongue[5]. The joystick had five different selections: up, down, left, right, and push. This technique also proved effective, but users quickly complained of tongue fatigue. Struijk encased loops of wire in a dental retiainer[9]. Users would then stick a metal core attached to the tongue in the loop. This action changed the inductance of the loop, which the system could detect. This system was less tiresome than the joystick method, but also much slower because users had to navigate the metal core into each loop.

The method proposed here uses infrared proximity sensors to detect the position of the tongue within the mouth. This technique offers several advantages: no external components need to be attached to the tongue, the tongue will not tire from pushing mechanical switches, and speed is not lost from having to navigate complex geometries. This paper investigates the feasibility of using the tongue as an input mechanism through the design of several computer programs that interface with the tongue input device.

Applications to ubiquitous computing

A tongue based input device serves the purposes of ubiquitous computing. With appropriate drivers, the device can serve as an input for any sort of electronics from computers to cars. The device is also harmonious with everyday life. The casual observer would not notice that a person is using the device to control other pieces of technology. The electronics of the device are embedded in a dental retainer, which is a widely used, fully established platform. The dental industry has been using retainers for decades. This gives the tongue based input device potential for expanding into a realm outside of assisted technologies. The electronics could be embedded in a retainer designed only for orthodontic purpose. The user, who has to wear the retainer to fix his or her teeth, could also control an ipod or a laptop computer.

Device Operation

The tongue based input device is pictured in figure 1. It consists of four infrared (IR) proximity sensors orientated in an up, down, left, right configuration. At this time, all of the signal processing is done outside of the retainer, so wires connect the proximity sensors to the control circuitry. These wires must hang out of a user's mouth. Future iterations of the project will include wireless communication.

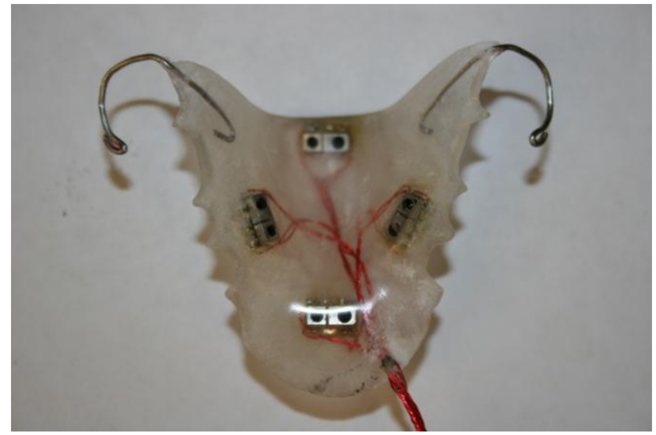

Figure 1: A complete tongue input device

The proximity sensors consist of an IR LED and an IR sensitive photodiode. Light emitted from the LED can reflect back into the photodiode as an object approaches the sensor. The intensity of light entering the photodiode creates a voltage swing across it. By calibrating the sensor, this voltage swing may be interpreted as a distance.

The proximity sensors are embedded in Ortho-Jet dental acrylic making the device completely bio-compatible. The stainless steel wires which hold the retainer in the mouth are positioned towards the rear of the mouth to prevent the retainer from being visually obvious.

The proximity sensors are connected to a PIC16F877 microcontroller, which handles control of the LEDs, reading the photodiode values through an analog to digital converter, and communicating with a remote computer. In a typical iteration of the program, the microcontroller will illuminate one LED on a particular sensor, read the corresponding photodiode voltage value, turn off the LED, and send the value to a remote computer for processing through a serial connection. Figure 2 shows a block diagram of the circuit.

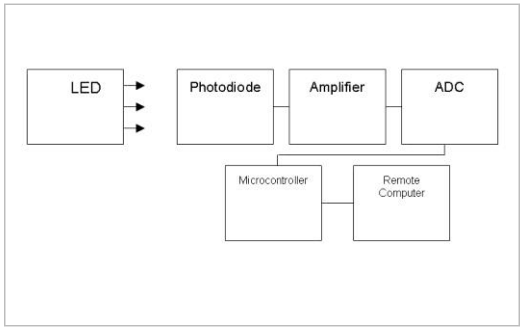

Figure 2: Block diagram of the tongue input device circuitry

Device Fabrication

Figure 3 shows the process steps for creating the tongue input device retainer. First a mold of the mouth must be taken. Alginate, a bio-compatible casting material derived from algae, is used for this step. Next, a plaster of paris casting of the mold is made. The retainer is built upon this casting so that it's shape fits the user's mouth. Medical grade stainless steel wires are bent to form clasps that hold the retainer in the mouth. These clasps must mimic the shape of the user's teeth for comfort. Next, a thin layer of acrylic is applied over the casting to provide an insulating layer between the sensors and the roof of the mouth. The sensors are then positioned in the casting. Dental acrylic is then built up around the sensors. Finally, the retainer must be ground to the rough shape and then sanded and polished to the final appearance.

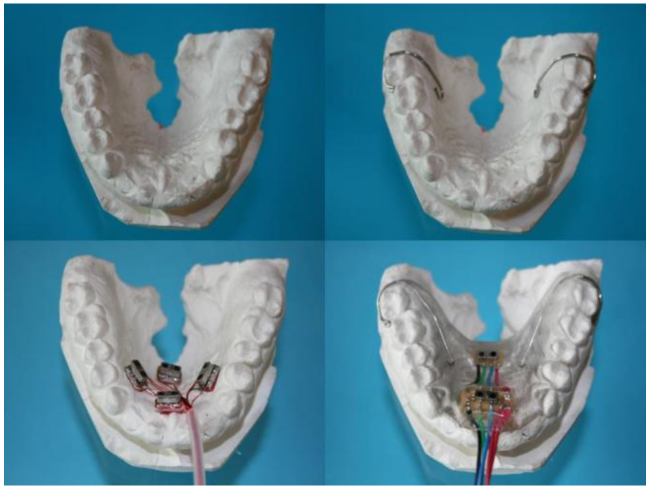

Figure 3: Process steps for fabricating a tongue input device

Software Description

We built two programs to illustrate that our device can be used both as a cursor and for text input.

To demonstrate cursor movement, we built a targeting game using Java and the Processing library that counted how many targets a user could mouse over in a period of time. Because our sensors only detected the tongue when they were <1mm away, each of the four sensors acted as a directional button. Also, the cursor could only move one direction at a time (direction determined by the maximum signal), which allowed for better control.

To illustrate keyboard input we implemented a text-input mechanism called EdgeWrite[4]. This technique

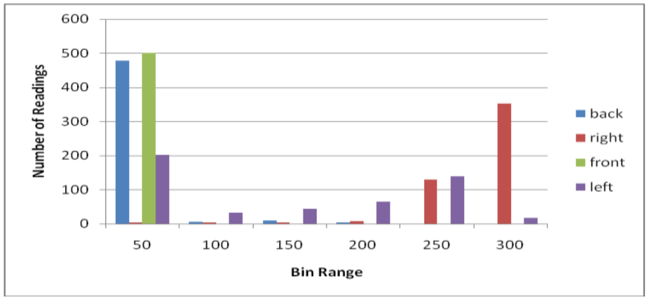

Figure 4 Touching right sensor.

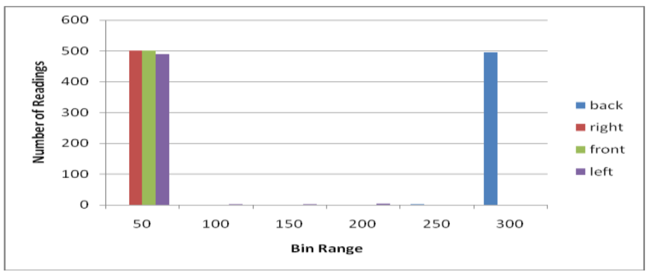

Figure 5 Touching left sensor.

has users input letters by creating gestures in one of four corners. We mapped each of the four sensors to a corner (rotated 45 degrees), and used the existing EdgeWrite API (in C#) to recognize gestures based on which sensors the tongue touched. Users would choose a letter by holding down the spacebar.

While there are many interesting games and demos we could have built, these two basic implementations illustrated that our device is capable of executing all essential input methods.

Figure 6 Touching the back sensor.

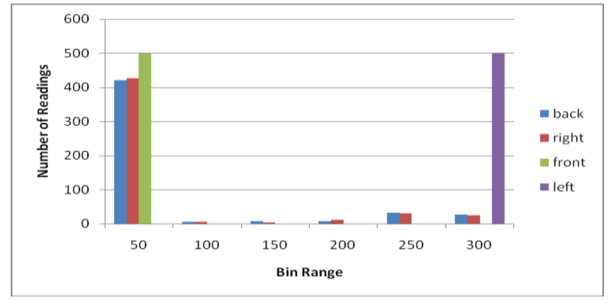

Figure 6 Touching left sensor.

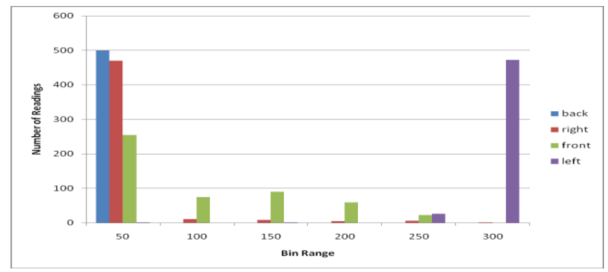

Figure 7 Touching front sensor.

Evaluation

Although our device enables users to execute both mouse and keyboard input, it will only be effective if it can be used efficiently. We evaluated our device in two ways. First, we examined the performance of the device: how well sensors are triggered for given movements. Second, we evaluated how much control tongue input allowed for cursor movement compared to mouse and keyboard input.

DEVICE PERFORMANCE

Our most basic evaluation looked at how well we could trigger individual sensors by touching them with our tongue. We did this by collecting 500 readings from each sensor while a user was doing a particular gesture, and evaluating whether the resulting readings behaved as we expected. Because we were assuming that our sensors acted like buttons, we expected that when we were touching a particular sensor, that sensor would be at its maximum (255), and all other sensors would be at their minimum (0). Figures 4,5,6,7 show histograms of readings for each of our 4 sensors.

From our data, we noticed that while the back sensor behaved as expected, none of the other sensors did.

When touching the left sensor (Fig 6), both the left and right sensors were at their maximums (though the left was maximized more often than the right). Touching the right sensor (Fig 4) produced similar results. This was most likely an artifact of the tongue's position in the mouth (the back of the tongue touched the left sensor while the tip touched the right) than a problem with the sensors. This presents a potential problem when considering tongue input, because we don't know of a way to distinguish between the tip and back of the tongue being near a sensor.

However, when touching the front sensor (Fig 7), the front sensor is never maximized, while the left sensor is. We think this is caused by two problems. First, the sensor was not well calibrated. We could make it either too sensitive (always maximized), or not sensitive enough (never maximized). Also, this illustrates that the sensor was very difficult to touch, suggestion that the front sensor should be placed farther back in the mouth.

CURSOR MOVEMENT

To evaluate how well our device could be used, we compared how well targeting tasks could be accomplished compared to a mouse and keyboard. We did this by counting how many times two expert mouse and keyboard users (the authors) could mouse over targets of different sizes in 30 seconds. The target size ranged from 60 to 10 pixels in diameter. We used a sample size of 10 taken from the same person for each input/radius combination, and the speed of the cursor was the same for both keyboard and tongue input.

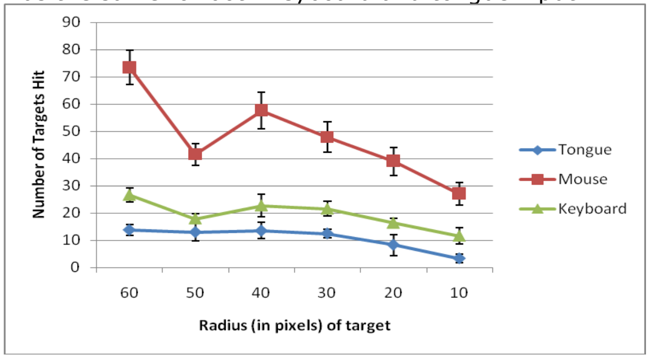

Figure 8 Targetting performance of mouse vs. keyboard vs. tongue. We used a sample size of 10 for each input/radius, and the error bars represent the standard deviation for each average value (each data point).

Fig 8 illustrates our results. As expected, our data shows that keyboard and mouse input are not equivalent (p-value 5×10^{-11}), and that keyboard and tongue input are not equivalent (p-value 2×10^{-12}). However, the tongue input was not much worse than keyboard input compared to the difference between mouse and keyboard input. The average difference in number of targets hit between keyboard and tongue input was 8.88, while the difference between mouse and keyboard input was 30.76. In other words, not much is lost by going from keyboard to tongue input.

Nevertheless, it is important to point out that while the keyboard and tongue inputs restricted the cursor speed, cursor speed on the mouse was not restricted. It would be interesting to explore how increasing cursor speed would affect the number of targets hit for tongue and keyboard inputs.

Problems and Sources of Error

There are several potential problems and sources of error to consider when examining our data, all of which we plan to eliminate in our future work.

First, the cursor targeting tests were done with a retainer that did not fit the user (due to time constraints). This is not a realistic depiction of how people would use the device, and therefore more studies need to be done with retainers that actually fit users.

Also, as mentioned before, our front sensor was not working properly, making it much harder to control the cursor initially. Finally, reflection from acrylic caused a noisy signal, which made it difficult to detect the tongue's distance from the retainer.

Drawbacks

The primary drawback to our idea is that we use a retainer. Retainers must be custom-fit to each person, and because of this are expensive to manufacture. This will greatly hamper how quickly these devices can be distributed. Furthermore, the cost of manufacture will limit who will be able to actually use these devices. However, this device is intended to be used for highly specialized purposes, where hand input is not possible or available. In these cases the needs will justify the cost.

Conclusion

Our initial prototype demonstrates the possibility of using infrared sensing as a convenient and feasible method for tongue-based input. We have shown that it is possible to take advantage of the tongue's dexterity to perform text and cursor input. While we would like our device to give more information on the tongue's position, even with our minimal implementation our device performs only slightly worse than a keyboard does for cursor input.

Future Work

We plan on improving the capabilities of our device by improvements in both hardware and software.

First, we hope to engineer a retainer which gives better information about the tongue's distance from the retainer. Second, we will make the front sensor easier to trigger by placing it farther back in the mouth. We also plan on embedding a Bluetooth radio in the retainer to run studies outside the lab.

These engineering improvements will make it much easier to accurately detect the tongue's position in the mouth. We hope to be able to not only detect the tongue's 3d position in the mouth, but also to use this to build applications for speech therapy and recognition. We also hope to great more natural tongue gestures (other than touching sensors) to control cursor movement so as to not fatigue the tongue.

Finally, we plan to equip our retainers with other sensors (i.e. visible light sensors) to give more context about a user's actions (i.e. whether they have opened their mouth).

Acknowledgements

We thank Shwetak Patel for teaching us prototyping techniques and Desney Tan for his idea

References

[1] Chen, Y. L., F. T. Tang, W. H. Chang, M. K. Wong, Y. Y. Shih, and T. S. Kuo. "The New Design of an Infrared-Controlled Human-Computer Interface for the Disabled." *IEEE TRANSACTIONS ON REHABILITATION ENGINEERING*. 7 (1999): 474-481.

[2] de Peralta, R. G., Q. Noirhomme, and S. L. Andino. "6. A High-Performance Brain Computer Interface Based on Visual Attention." *CLINICAL NEUROPHYSIOLOGY -SHANNON-.* 119. 9 (2008): e166-e167.

[3] Huo, X., Wang, J., & Ghovanloo, M. (2007). A Wireless Tongue-Computer Interface Using Stereo Differential Magnetic Field Measurement. IEEE Engineering in Medicine and Biology Magazine : the Quarterly Magazine of the Engineering in Medicine & Biology Society. 1, 5724.

[4] Nutt, W., Arlanch, C., Nigg, S., & Staufert, G. (1998). Tongue-mouse for quadriplegics. Journal of Micromechanics and Microengineering : Structures, Devices, and Systems. 8 (2), 155.

[5] Qiyu, P., & Budinger, T. (2007). ZigBee-Based Wireless Intra-Oral Control System for Quadriplegic Patients. IEEE Engineering in Medicine and Biology Magazine : the Quarterly Magazine of the Engineering in Medicine & Biology Society. 1, 1647.

[6] Ramdane-Cherif, Z., and A. Nait-Ali. "An Adaptive Algorithm for Eye-Gaze-Tracking-Device Calibration." *IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT*. 57. 4 (2008): 716-723.

[7] Santhanam, G., S. I. Ryu, B. M. Yu, A. Afshar, and K. V. Shenoy. "A High-Performance Brain-Computer Interface." *NATURE -LONDON-*. 442. 7099 (2006): 195-198.

[8] Shih SW, and J Liu. "A Novel Approach to 3-D Gaze Tracking Using Stereo Cameras." *IEEE Transactions on Systems, Man, and Cybernetics. Part B, Cybernetics : a Publication of the IEEE Systems, Man, and Cybernetics Society*. 34. 1 (2004): 234-45.

[9] Struijk, L. N. S. A. "An Inductive Tongue Computer Interface for Control of Computers and Assistive Devices." *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING BME*. 53. 12 (2006): 2594-2597.

[10] Takami, O., K. Morimoto, T. Ochiai, and T. Ishimatsu. "Computer Interface to Use Head and Eyeball Movement for Handicapped People." *IEEE INTERNATIONAL CONFERENCE ON SYSTEMS MAN AND CYBERNETICS*. 2 (1995): 1119-1123.

[11] Wobbrock, J. O., Myers, B. A., and Kembel, J. A. 2003. EdgeWrite: a stylus-based text entry method designed for high accuracy and stability of motion. In Proceedings of the 16th Annual ACM Symposium on User interface Software and Technology (Vancouver, Canada, November 02 - 05, 2003). UIST '03. ACM, New York, NY, 61-70. DOI= http://doi.acm.org/10.1145/964696.964703

[12] Zhu, Z., and Q. Ji. "Novel Eye Gaze Tracking Techniques Under Natural Head Movement." *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING BME*. 54. 12 (2007): 2246-2260.