TeethClick: Input with Teeth Clacks

Tamer Mohamed Lin Zhong
Dept. of Electrical & Computer Engineering
Rice University, Houston, TX 77005
{lzhong}@rice.edu

ABSTRACT

We present *TeethClick*, a simple yet robust input technique based on automatic detection of teeth clacks with a throat microphone. We address its user interface design and provide results from user studies. TeethClick is intended as an auxiliary input method for hand-free computer operation, especially for people with motor impairments. Compared to alternative technologies, TeethClick provides a novel low-cost, accurate, noise-resilient, and user-friendly solution that incurs very little computation load. Our user studies indicate that subjects learned TeethClick and achieved decent performance quickly.

INTRODUCTION

When we speak, the upper jaw can touch the lower jaw, producing teeth clacks accidentally. Teeth clacks have been considered detrimental to speech recognition [1]. Techniques were proposed to eliminate them from jaw-bone microphone signal for noise-resilient speech recognition [1]. In this work, we propose to do the opposite: reject speech and detect gentle yet deliberate teeth clacks for computer input. We call this technique *TeethClick*. TeethClick employs a low-cost throat microphone to pick up bone-conduction signal from the cheek. It uses very efficient spectrum analysis to distinguish teeth clacks from other vocal activities. Using TeethClick, users with motor impairments can easily generate accurate computer input. We intend TeethClick as an auxiliary and complementary input technique to more versatile input techniques, such as speech recognition.

The paper is organized as follows. We first present TeethClick and then provide some example user interface designs. After that, we discuss our user studies as well as address related work.

TEETHCLICK

TeecthClick uses a single throat microphone that touches the cheek and picks up the vibration signal from the jawbone. We call this signal “bone-conduction” in this work. Our implementation and analysis are based on the throat microphone from a Noise Terminator headset from IASUS Concepts [2].

Property of Bone-Conduction Signal

The bone-conduction signal of teeth clacks is characterized by high energy in spectrum above 2000Hz but low energy below it. Figure 1 shows the time-spectrum of the bone-conduction signal of several teeth clacks. The spectrum of the bone-conduction signal of speech, as shown in Figure 2, is almost the opposite. It is characterized by high energy in spectrum below 2000Hz but low energy above it. This dramatic difference is introduced by that the skin and skull is a much lower-pass filer to acoustic signals than to bone-vibration incurred by teeth clacks. This forms the basis for our algorithm to detect teeth clacks.

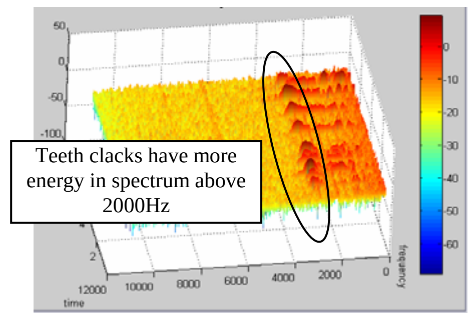

Figure 1. Spectrum vs. time for bone-conduction signal of a series of deliberate teeth clacks

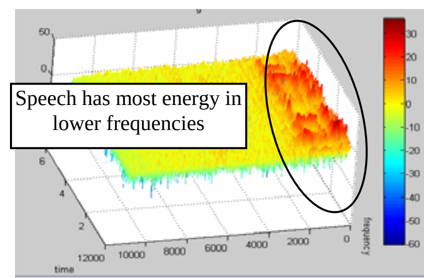

Figure 2. Spectrum vs. time for bone-conduction signal of speech

Detecting Deliberate Teeth Clacks

For low-power and real-time implementation, we design a simple-yet-effective algorithm based on the property of the bone-conduction signal. The algorithm simply examines the energy densities in the lower and higher spectral ranges of the bone-conduction signal. High energy density in the lower spectral range indicates the existence of speech, while a sudden increase in the energy density in the higher spectral range indicates the occurrence of a teeth clack. A deliberate teeth clack is detected if a teeth clack occurs without the presence of speech.

Our implementation is based on standard speech signal processing: We sample the bone-conduction signal and divide the samples into overlapping frames. In our implementation, each frame is about 23.3ms and adjacent frames are about 22ms apart. For each frame, we conduct FFT to get its spectrum. In our implementation, the “low” spectral range is between 0 and 2750Hz, while the “high” spectral range is between 1875 and 5500Hz. Through experiments, we discovered that such overlapping ranges work best. For the *n*th frame, we calculate the energy densities in the low and high spectral ranges, denoted as A_n and B_n, respectively. We also keep a record of the average energy density of silence, U.

If B_n is considerably larger than B_{n-1} and B_{n+1}, the algorithm declares that a teeth clack is detected. For accidental teeth clacks, A_{n-1} and A_{n+1} are large due to the presence of speech. Therefore, the algorithm declares that a deliberate teeth clack is detected if and only if B_n is considerably larger than B_{n-1} and B_{n+1} AND A_{n-1} and A_{n+1} is on the same level as the U. Let C_n be the Boolean logic that evaluates whether a deliberate teeth clack is detected for the nth frame. It can be formulated as

$$C_n = [(B_{n-1} + offset) < B_n] \text{ and } [(B_{n+1} + offset) < B_n] \\ \text{and } [A_{n-1} \le (U + offset)] \text{ and } [A_{n+1} \le (U + offset)]$$

where *offset* is empirically set to 5dB. It is important to note that while the algorithm is based on the generic property of the bone-conduction signal, its implementation is highly dependent on the property of the throat microphone. In our implementation, the low and high spectral ranges as well as the offset were empirically determined by examining the bone-conduction spectrum.

Producing Multiple Inputs

Teeth clacks must generate different inputs for a user to operate a computer. We have investigated two ways to produce different inputs.

Multiple clicks: If we view a teeth clack as a tact button push, we can have different inputs for consecutive multiple clacks. To distinguish between a "single click" and a "double click", we continuously analyze a first-in first-out (FIFO) buffer that stores C_n for the most recent frames within 400ms. If a teeth clack is detected and no teeth clack is detected in following 300ms, we treat it as a *single click*. If two consecutive teeth clacks are separated by less than 100ms, we treat them as the one clack and use earlier time as its on-set time. If two consecutive clacks are separated by more than 100ms and less than 300ms, we treat them as a *double click*. The rate of single clicks is limited by the 300ms delay for every single click that we have to wait to tell whether it is part of a double click.

Single clicks are faster and require less effort than double-clicks; they should be used for more frequent tasks. This is similar to the philosophy of Huffman coding, which uses shorter code words for more frequent symbols.

Our tests showed that this algorithm provides reliable detection of single and double clicks. Extending it to triple clacks will require user-specific calibration and introduces further latencies in single-click detection. Therefore, TeethClick only uses single and double clicks, effective implementing a tact button with teeth clacks.

Teeth clack location: We considered recognizing clacks at different locations of the jaw, using two throat microphones placed on both sides of the cheek. We found that recognizing the location is difficult and unreliable, because it requires stereo processing and multiple microphones. It is very sensitive to the positions of microphones. Moreover, producing located teeth clacks requires much higher physical effort and can easily introduce facial muscle strain. Therefore, we choose not to use teeth clack location.

USER INTERFACE DESIGN WITH TEETHCLIK

TeethClick is functionally the same as a mouse button. We intend it as an auxiliary input technique to other more versatile input technologies. However, TeethClick enjoys the advantage as being hand-free and non-intrusive. By carefully designing the user interface, TeethClick can be used by people with motor impairments. In this Section, we present two examples that implement two basic GUI operations: selection and pointing. Selection refers to choose one item from a list. Pointing refers to move the cursor to a certain position inside a window.

Selection: To select from a list of items, a single click highlights the next item, which is a more common and often repeated task. A double click selects the current highlighted item.

Pointing: We can build most GUI operations based on the selection operation. A user can use TeethClick to move the cursor to any block with two actions, "Rotate" and "Move". "Rotate" changes the cursor orientation. "Move" moves the cursor to the next block along the direction it is oriented. Since "Move" is more frequent than "Rotate", we could use single clicks to "Move" and double clicks to "Rotate." Since "Move" is highly repeated, we use a double click to switch between "Move" and "Rotate" and a single click to repeat the chosen action.

USER STUDIES AND EXPERIMENTAL RESULTS

We implemented TeethClick using the MATLAB data acquisition toolbox. We recruited four subjects, two male and two female ECE graduate students, to participate in user studies for the selection and pointing operations with TeethClick.

Experiments

We explained TeethClick as well as the selection and pointing operations to subjects at the beginning. We then allowed them five minutes to play with selection and pointing each. Then we started measuring their performance for selection and pointing tasks. We carried out selection experiments before pointing ones. We measured multiple trials to see the learning curve in a short time.

Task 1: Selection

The first task is to use TeethClick to select numbered items from a menu in an arbitrary but predefined sequence. The menu includes only a sequence of numbers, from 1 to 6, to minimize distraction. The subject was asked to select the numbers in the order of 3,5,6,4,1,5,2. This task was carried three times with five-minute break in between. The total time to finish each trial is shown in Figure 3. The user performance in the second trial was significantly better than the first one. The primary reason was that subjects made much fewer errors in the second trial. However, the performance slightly deteriorated in the third trial. According to our post experiment interviews, this was due to that the subjects became overconfident after two trials when they realized how easy TeethClick was and became less focused.

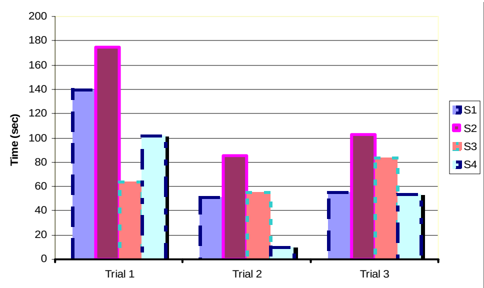

Figure 3. Performance for the selection task

Task 2: Pointing

In the second task, the subject was asked to move a cursor-like pointer in a GUI with the design discussed in the above. The GUI is shown in Figure 4. Twelve objects were drawn on the grid clockwise. The subject used TeethClick to move the pointer to select a series of objects in the order of 3, 9, 10, 2, 7, 6 and 5. The total time to complete this series of pointing tasks is shown in Figure 5 for three trials. The data for the fourth subject in the second trial is not shown because the subject was disturbed during the experiment. There is less improvement for the three trials as compared to that in Task 1. This is because experiments for Task 2 were conducted later.

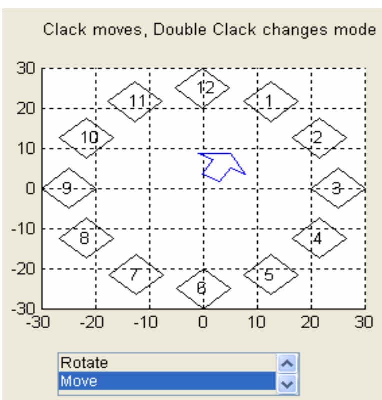

Figure 4. A cursor pointing task using a list of two actions: "Rotate" and "Move"

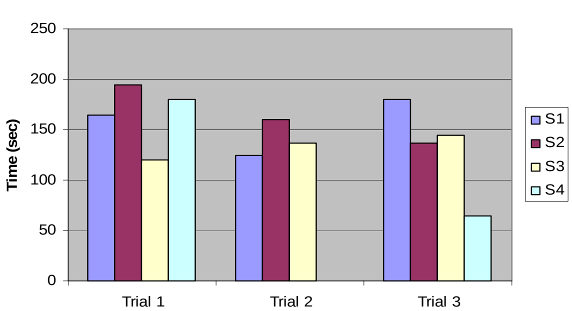

Figure 5. Performance for the pointing task

User Evaluations

After a subject finished the experiments, we asked her/him to fill a questionnaire for a subjective evaluation. We also interviewed the subjects for questions we had with their behavior in the experiments and their answers in the questionnaire. We summarize our findings as follows.

First, although our subjects do not have motor impairments, all of them indicated that they had encountered situations that require hand-free computer operation. They all recommend TeethClick to people with difficulty using hands. The average score is 6.25 as 1 being "Not recommend at all" and 7 being "Strongly recommend". All of them indicated that they would like to have TeethClick if they could not use their hands for a whole week.

Second, we asked subjects how comfortable they feel with using TeethClick in the experiments. The average level of comfort is 3.5 with 1 being "Extremely uncomfortable" and 7 being "No problem at all". There was a concern of jaw strain after elongated usage. This concern was unexpected because TeethClick requires very gentle teeth clacks with little facial muscle effort. However, we observed that all our subjects made strenuous teeth clacks in the experiments, which introduced discomfort after 45 minutes of TeethClick usage, despite that we told them gentle clacks work as well. We found out from interviews that they unconsciously assumed that a forceful clack could be more accurate. They also felt in better control when forceful clacks were used. Since it was the first time our subjects had used TeethClick and they had only about 45 minutes of experience with it, we believe these issues will become less a problem after enough exposure to TeethClick. However, a more extensive and longer term user study is necessary to investigate them. Our subjects indicated a 6.5 average level of comfort for wearing the throat microphone, based on the same 1-7 scale.

Third, the average subject satisfaction score for TeethClick speed is 4.25, as 1 being "Very disappointed" and 7 being "Very satisfied". Our subjects also noticed that the GUIs used in experiments can be improved in many aspects for higher performance. This highlights the importance in designing user interfaces based on TeethClick. Another note is that our measurement showed that users can easily make 4 to 5 teeth clacks per second, a speed comparable to mouse button clicks. This indicates that TeethClick has a lot of room for improvement.

Fourth, we asked subjects their opinions on speech recognition vs. TeethClick. For hand-free computer operations, three of them would like to have both speech recognition and TeethClick. There was a concern of privacy for using speech recognition in office environments. They all believe that speech recognition is the best available technology for hand-free computer operation, although only one of them had ever used speech recognition. This is not surprising at all since speech recognition has received more than 20 years of research and has been marketed by major companies, including Microsoft and IBM, for many years. We believe that TeethClick complements speech recognition and has a lot potential to serve as a hand-free auxiliary input technique.

To summarize, TeethClick is a new technique to users and many issues remain to be addressed. The most important issue is to ensure user confidence in using gentle teeth clacks. Our subjects did recognize that TeethClick has its own value and recommend it for people who need hand-free computer operations.

RELATED WORK

Input techniques similar to TeethClick were explored in the past, especially for people with motor impairments. Some used the tongue and in-mouth springs, switches, or joysticks. For example, Kingma and Sabourin developed a "mouth-mouse" for quadriplegic computer users [3]. A mouth-mouse user uses the tongue to push several in-mouth springs for mouse moving and bites a switch for mouse clicking. Similar tongue-operated in-mouth input devices were reported in [4,5]. TeethClick is much more hygienic, less intrusive, and easier to operate. Speech recognition has been well studied as an input method for people with motor impairments [6]. Non-speech voice was employed in [7, 8]. TeethClick is not intended to replace these voice-based techniques. Instead, it provides a much more efficient and low-cost auxiliary method. Since TeethClick uses the bone-conduction signal, which is almost immune from environmental noise [1], it is much more noise-resilient than speech recognition, which uses close-talk microphones.

DISCUSSION

Dental health issue: By the time of this writing, we are not able to find literature indicating that teeth clacks have a negative effect since they occur naturally during talking. Since only gentle teeth clacks are required, we would believe the detrimental effect, if any, would be limited and only for extensive usage. For example, we do not expect temporomandibular dysfunction. Nevertheless, we believe that the dental health issue needs to be investigated as a future work.

Importance of interface design: We would like to emphasize that TeethClick is only an input technique. User performance with TeethClick very much depends on the user interface design, as we found out in the user studies. Although we used two GUI mouse operations, selection and pointing, as examples, we believe TeethClick works better as an auxiliary input method instead of the mouse replacement. For example, TeethClick can be used to operate consumer electronics, such as Bluetooth headsets, due to its simplicity and efficiency. These devices have much limited user interfaces that may work better for TeethClick. Moreover, although we presented TeethClick as a simple and elegant new input technique, it can be readily combined with other input technologies.

CONCLUSIONS

We presented the design and user studies of TeethClick, an input technique based on robust detection of deliberate gentle teeth clacks. Combined with proper user interface designs, TeethClick can be used to operate computers hand-freely. Our user studies showed that TeethClick is easy to learn and achieve decent performance even in GUI-mouse tasks, such as selection and cursor pointing. However, as a fresh technique, TeethClick still faces many challenges, especially in user interface design for minimal discomfort and maximal performance. Its impact on dental health needs investigation too. While speech recognition has a significant establishment, we believe TeethClick offers one more choice for hand-free computer and consumer electronics operations. Compared to speech recognition, its cost is low, its implementation is simple, it is noise-resilient, and it is free of privacy concerns. All these make it an ideal complementary/auxiliary technique to speech recognition.

ACKNOWLEDGMENTS

We would like to thank our volunteers for participating in the user studies.

REFERENCES

1. Zicheng Liu et al, "Leakage model and teeth clack removal for air-and-bone conductive integrated microphones", Proc. *IEEE Int. Conf. Acoustics, Speech & Signal Processing (ICASSP)*, Philadelphia, 2005.
2. IASUS Concepts, Noise Terminator (NT): http://www.iasus-concepts.com/noise_terminator/
3. Y. J. Kingma and P. J. Sabourin, "Development of a mouth-mouse for quadriplegics," in Proc. *IEEE Ann. Int. Conf. Engineering in Medicine and Biology*, pp.1516-1517, Nov., Seattle, WA, 1989.
4. C. Salem and S. Zhai, "An isometric tongue pointing device," In *Proc. SIGCHI Conf. Human Factors in Computing Systems*, pp. 538-539, Atlanta, GA, March 22 - 27, 1997.
5. Dongshin Kim, Mitchell E. Tyler, David J. Beebe, "Development of a Tongue-Operated Switch Array as an Alternative Input Device, " *Int. J. Human-Computer Interaction*, 18:1, 19-38, 2005
6. Bill Manaris, Valanne MacGyvers, and Michail Lagoudakis, "A Listening keyboard for users with motor impairments-a usability study, " *Int. J. Speech Technology*, Dec. 2002, pp. 371-388.
7. Takeo Igarashi and John F. Hughes "Voice as sound: Using non-verbal voice input for interactive control " Proc. *UIST* 2001, pp. 155-156.
8. Jeff A. Bilmes et al, "The Vocal Joystick: A voice-based human-computer interface for individuals with motor impairments," Proc. *Human Language Technology Conf. & Conf. Empirical Methods in Natural Language Processing* (HLT/EMNLP), pp. 995--1002, Vancouver, Canada, Oct. 2005.