A Voice-Based Input System for Embedded Applications with Constrained Operating Conditions

Harshit Bangar
Department of Electronics and
Electrical Engineering
Indian Institute of Technology
Guwahati
Guwahati 781039, India
h.bangar@iitg.ernet.in

Maulishree Pandey
Department of Design
Indian Institute of Technology
Guwahati
Guwahati 781039, India
maulishree@iitg.ernet.in

Dhruv Kapoor
Department of Electronics and
Electrical Engineering
Indian Institute of Technology
Guwahati
Guwahati 781039, India
k.dhruv@iitg.ernet.in

Pradeep G. Yammiyavar
Department of Design
Indian Institute of Technology
Guwahati
Guwahati 781039, India
pradeep@iitg.ernet.in

CSCW '13 Companion, Feb. 23–27, 2013, San Antonio, Texas, USA.
ACM 978-1-4503-1332-2/13/02.

Abstract

This paper presents a voice-based input mechanism for embedded applications with limited computing power and requiring only a small set of inputs, but with the user constrained in not being able to user her or his hands. The requirement of designing for embedded systems, which places a strict upper limit on available computing power, in addition to the restriction on haptic feedback, makes designing a suitable interface a unique challenge. The proposed solution relies on an ingenious extension of Voice Activity Detection providing inherent disturbance immunity, and incorporates feedback to the user to make selection of the input easier. This input scheme is being tested for a unique system for cleansing immobile patients via nozzle-fitted gloves worn by an operator on both hands. The design calls for the operator being able to control the flow of soap, water, or a stream of hot air from the nozzles, even with both hands engaged in cleaning the patient using the gloves.

Introduction

While haptic control is by far the most common mode of providing input to a device, a number of scenarios do not allow for any haptic feedback, especially where the hands may be occupied in operating equipment or machinery, presenting a challenge for the design of a suitable input mechanism for controlling the equipment. The problem is exacerbated by the lack of significant computing power in applications driven by simple embedded systems. Implementing an input mechanism for embedded devices poses its own unique set of challenges owing to the limited computing power and parallel processing ability, the lack of complex operating systems, and other limitations associated with such systems.

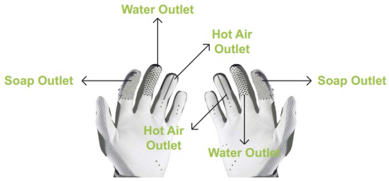


Figure 1: A diagram showing the nozzle-fitted gloves for the patient cleaning system.

As an illustration of a typical scenario where a non-haptic feedback method would be advantageous, and which calls for using low-end microcontrollers to keep costs low, is a novel system under development at IIT Guwahati that enables nurses or attendants to cleaning immobile patients in a quick, practical and sanitary fashion. The device relies on a pair of gloves fitted with nozzles, as depicted in Fig. 1 that can route either soap or water from reservoirs in the moveable unit to clean the patient, and can also direct hot air to finally dry of the patient's skin.

The voice-based input system presented in this paper was developed originally for the patient cleaning system described above. The requirement for the operator to use both hands to clean the patient with the nozzle-fitted gloves, which may often be spraying water or soap, making moving them off the patient cumbersome, necessitated having an entirely non-haptic input mechanism for controlling the flow of fluid, selecting between water, soap, or hot air, and other possible control inputs. The likely motion of the operator along the patient's body made using foot-pedals or some form of rudimentary gesture recognition a sub-optimal solution, leaving speech-based control as the alternative.

However, conventional speech-based control replying on interpreting spoken words has a number of significant drawbacks for an application such as this, where keeping costs should be low and therefore using full-fledged computers or servers and accompanying wireless networks are to be strongly avoided. Most speech-recognition tasks are computationally heavy and therefore speech recognition on mobile devices such as Apple's Siri [1] is generally performed by sending speech data to a central server over the internet. This is impractical for this application because implementing a networked solution would add significantly to cost, and because the application requires *always on* voice recognition, since the operator may provide a control input at any time, meaning significant network usage and the necessity of dedicated servers for each machine. Additionally, conventional speech based solutions have to contend with ambient noise [2, 3] which may be quite high in a setting like a hospital. Lastly, there would be possible interference caused due to the patient and the operator conversing with each other, which may result in false triggers to the control system. All these disadvantages add to the significant difficulty inherent in designing systems that can recognize words from human speech, including the large amount of training data needed, and the possible need for additional training data from individual operators of the machine.

For this application, a highly modified form of Voice Activity Detection, in conjunction with aural feedback and other aides to help the operator select the right input, was developed. The developed method, described in the next section, is highly immune to noise, simple to use, requires almost no tuning at all, and can be easily implemented on embedded devices.

The Proposed Voice-Based Feedback-Aided Input System

A suitable input system for an embedded application must be robust to noise, low in computational complexity, have

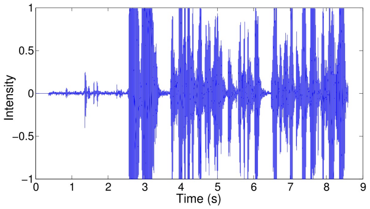


Figure 2: A speech signal of a person speaking relatively fast. Note that there are significant gaps between words, which would by and large be the peaks seen in the graph.

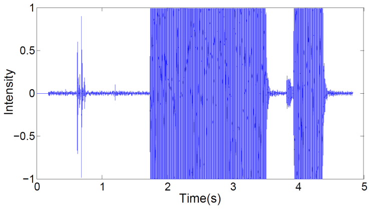


Figure 3: The signal made when the user provides the input 1101. Notice how different the signal looks when the user *hums*, as compared to the regular speech signal shown in Fig. 2

minimal hardware requirements, and be easy to operate. As explained earlier, convectional speech recognition would be entirely unsuitable for low-cost embedded applications where significant disturbance is to be expected.

Voice Activity Detection is a relatively simple speech processing task used to detect the presence or absence of speech signals above a certain threshold, or containing at least a certain amount of *energy* in a given time period. This decision can be taken quite easily with a heuristically tuned measurements that counts the number of samples of a fixed sample-rate speech signal that are greater than a certain threshold against a fixed value. The threshold can be tuned based on the sensitivity of the mike, and voice activity can be said to be detected if the number of samples measuring greater than the threshold exceed the fixed value. When this fixed value is chosen to be low, even very minor disturbances, such as those affected in the course of normal speech, would activate the detector. However, in the course of normal speech, there are significant silent gaps between, or even within, words, as Fig. 2 demonstrates. Therefore, with a higher value, it can be made almost certain that no everyday speech would trigger the detector, so that for a successful trigger, the operator would, in effect, need to *hum* to send in a lot of high intensity signals to the system.

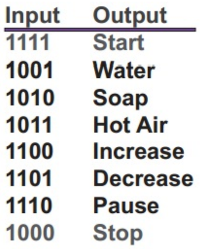
While *humming* can be an effective way to register an input, it is severely limited in the number of inputs that can be deciphered. Nevertheless, using this variation of Voice Activity Detection as the basic building block of the final input mechanism provides several advantages, including high noise immunity, the lack of any need to tune behaviour for each operator, and adaptability to embedded devices.

A natural extension of the idea of using humming to provide an input would be checking for the duration the user has hummed. This in turn could be guided by feedback from the machine using clearly audible *beeps*, which would sound off after, say, every one second. This extends the rudimentary idea explained above and allows for multiple inputs, but is an extremely cumbersome solution wherever the operator might be expected to provide more than three or four inputs.

In order to allow for a large number of inputs to be provided, the input can be given as a sequence of binary digits. Silence during a fixed interval of time can be coded as a 0, while humming can be coded as a 1. This basic idea requires making a few considerations for the duration of the time interval, the minimum period needed for humming to register as a 1, providing aural feedback to indicate completion of part of the input, etc. Fig. 3 shows an example of this type of input scheme. The details are described below.

Algorithm for the Proposed Input System
The use of a highly modified form of Voice Activity Detection in conjunction with aural feedback to provide binary strings as inputs allows for choosing between a large number of possible inputs in a short period of time. To allow for the input to be provided at a fast pace, it is important that the period for each input bit be kept at a minimum. This, however, brings up the risk of accidental triggering from regular speech.

To avoid this, a *trigger bit* of a longer duration is compulsorily required before any input. Thus, to provide an input, say, 011, the correct sequence would be $1 - 011$, with the first 1 being the trigger bit which is to be hummed for a longer time. After the trigger bit, and after any regular input bit, a beep will be sounded to indicate



Input	Output
1111	Start
1001	Water
1010	Soap
1011	Hot Air
1100	Increase
1101	Decrease
1110	Pause
1000	Stop

Figure 4: A sample input chart for the patient cleaning system.

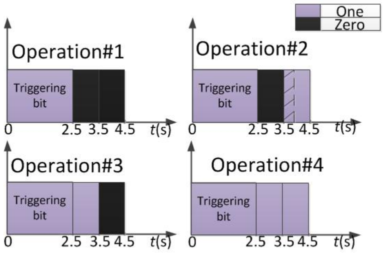


Figure 5: Operations 1 through 4 represent four different inputs: 100, 101, 110, and 111. For registering a one, a relaxation is provided so that the user only needs to hum in the last .5 seconds, rather than the entire duration of 1 second (note that this does not apply to the trigger bit). This is necessary in cases such as operation 2. While operations 1, 3, and 4 are straightforward, operation 2 requires the user to start humming again after stopping for the second bit. This relaxation is need since the user cannot start humming immediately after the beep that indicates the 0.

that the users signal has been registered. This would let the user know, firstly, that the trigger bit has been registered, and later that an input bit has been entered and the next input bit is to be provided in case the input isn't already complete. To facilitate providing input signals, the operator may use a simple chart mapping operations to their corresponding binary codes. A sample chart for the patient cleaning system is shown in Fig. 4.

As explained earlier, for any input to register as a 1, the user must hum for the most part of the input period. The algorithm ensures this is so by counting the number of samples whose intensity is above a threshold that can be selected based on the mic's sensitivity – these are the 1-samples). If the number of 1 samples exceeds a certain fixed value, the input bit would be registered as a 1, and otherwise as a 0.

Figure 5 displays the user inputs expected for a 2-bit system. Note that an initial triggering bit is needed, as can be seen with all possible input combinations in Figure 5, so as to indicated beginning of input. The triggering bit would need to be provided for a longer duration, so as to be clearly distinguishable from regular speech. In this particular example, the trigger requires the user to hum for $2.5s$, while the actual input bits require only $1s$ of input. Beeps after the trigger and the input bits indicate to the user that a particular trigger or input has been registered, and that he or she may proceed to the next bit in case the input is yet to be completed. An important design element is the relaxation of the requirement for a humming for all non-triggering bits. Humming is needed only in the last half of the $1s$ period for each input bit. This is done since while switching from providing a 0 bit to providing a 1 bit, the user would inevitably take some time to start humming again.

User-Survey and On-Going Work

A user-survey was carried out among 6 individuals, none of whom had used the system before. To ensure that regular conversation would not trigger the system, the users were explained the procedure while the system was running. No accidental triggers were detected. Since ease of learning was of primary importance, the users were given only a minute to play with the input system, and then were presented with a series of inputs they were supposed to provide to the system. 100% of users were able to provide the inputs 1000, 1001, and 1010, while the number was 83.3% for the remainder of the input combinations. It was seen that with a little bit more practice, users were able to hit the right input almost every time.

The method proposed in this paper is planned to be incorporated into the patient cleaning system described earlier in the paper. It is hoped that using this input scheme will allow operators of the machine to use both hands on the patient, while still being able to control the behaviour of the system. The easy learning curve should mean that little training would be required to familiarize operators with the input scheme.

References

[1] Wiping Away Your Siri "Fingerprint", MIT Technology Review. http://www.technologyreview.com/news/428053/wiping-away-your-siri-fingerprint/.

[2] Gong, Y. Speech recognition in noisy environments: A survey. *Speech Communication 16*, 3 (April 1995), 261–291.

[3] Singh, R., Seltzer, M., Raj, B., and Stern, R. Speech in Noisy Environments: Robust Automatic Segmentation, Feature Extraction, and Hypothesis Combination. In *IEEE Conference on Acoustics, Speech and Signal Processing* (May 2001).