Voice as Sound: Using Non-verbal Voice Input for Interactive Control

Takeo Igarashi *John F. Hughes*
Computer Science Department, Brown University
115 Waterman Street, Providence, RI 02912, USA
takeo@acm.org, jfh@cs.brown.edu

ABSTRACT
We describe the use of non-verbal features in voice for direct control of interactive applications. Traditional speech recognition interfaces are based on an indirect, conversational model. First the user gives a direction and then the system performs certain operation. Our goal is to achieve more direct, immediate interaction like using a button or joystick by using lower-level features of voice such as pitch and volume. We are developing several prototype interaction techniques based on this idea, such as "control by continuous voice", "rate-based parameter control by pitch," and "discrete parameter control by tonguing." We have implemented several prototype systems, and they suggest that voice-as-sound techniques can enhance traditional voice recognition approach.

KEYWORDS: Voice, Interaction technique, direct manipulation, entertainment.

INTRODUCTION
Typical voice-based interfaces focus primarily on the verbal aspects of human speech. Speech recognition engine turns speech into words or sentences, and the system performs appropriate actions based on recognized texts. One of the limitations of these approaches is that the interaction turnaround is long. The user must complete a word and wait for the recognition results. While this is reasonable for complicated tasks like flight reservation, it is inappropriate for direct, low-level controls such as scrolling. This paper proposes the use of non-verbal features in speech, features like pitch, volume, and continuation, to directly control interactive applications.

RELATED WORK
Speech recognition researchers have started using non-verbal, prosodic features in speech to increase the accuracy of traditional speech recognition and semantic language processing [2][4][5]. Some conversational systems also use non-verbal information to enhance interaction. Tsukahara and Ward described an electronic tutor system that detects the user's emotional state from the prosody of her utterances, and adjusts its behavior accordingly [7]. Goto *et. al.* described a voice-completion interface [1] that detects each filled pause during an utterance as a trigger and automatically completes the utterance like tab-key completing commands in Unix shells.

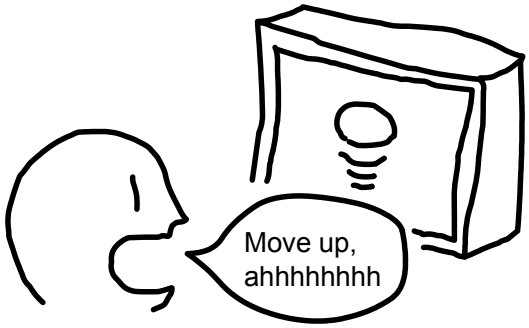

Figure 1. The user controls the application directly using continuous voice, and the system provides immediate feedback.

The SUITEKey system is a speech interface for controlling a virtual keyboard and mouse for motor-disabled users [6]. When the user says "move mouse down … stop", the mouse pointer moves downward during the pause. Our techniques extend this work by introducing additional interaction techniques for voice-based direct manipulation.

INTERACTION TECHNIQUES

Control by Continuous Voice
In this interface, the user's voice works as an on/off button. When the user is continuously producing vocal sound, the system responds as if the button is being pressed. When the user stops the sound, the system recognizes that the button is released. For example, one can say "Volume up, ahhhhhh", and the volume of a TV set continues to increases while the "ahhh" continues. The advantage of this technique compared with traditional approach of saying "Volume up twenty" or something is that the user can continuously observes the immediate feedback during the interaction. One can also use voiceless, breathed sound.

Rate-based Parameter Control by Pitch
This technique extends the previous technique by allowing the user to adjust an additional parameter by pitch while continuously producing sound. This technique works as a

one-dimensional joystick, lever, or a slider. An example is map navigation. When the user says “move up, ahhhh”, the map on the screen scrolls down while the sound continues. When the user increases the pitch of his voice, the scrolling speed increases, and vice versa. When the user stops speaking, the scrolling ends. We also combined this technique with a speed-dependent automatic zooming interface [3].

Discrete Control by Tonguing

The preceding techniques are suitable for continuous parameter control. By contrast, for discrete value selection, we use “tonguing.” For example, when the user says “Channel up, ta ta ta,” the channel number increases by three. Since this technique simply detects discrete peaks in sound signal, one can also use bodily actions such as hand clapping and finger snapping to make noise.

IMPLEMENTATION

Our prototype systems are implemented in C++ and Java on Windows. The low-level signal processing part is written in C to calculate the voice spectrum, and Java code performs high-level processing based on the spectrum. In our current implementations the input signal is digitized at 16 bit / 22 kHz, and then the short-time Fourier transform (STFT) with a 2024-sample Hanning window is calculated by using the Fast Fourier Transformation (FFT). The FFT frame is shifted by 256 samples, and the discrete time step is 12ms.

Voice detection is achieved by checking the total volume (dB) of the input spectrum. The low-frequency part (<375Hz) of the spectrum is removed to reduce the effect of background noise. Pitch transitions (up or down) are detected by comparing the dot products between the time-shifted (12ms), frequency-shifted (±43Hz) voice spectra. Note that our algorithm does not calculate the absolute pitch. Tonguing is detected by counting short voiced regions in input sequence. Detecting voice and tonguing is fairly robust among several test users, but pitch detection does not work well for some users.

The speech recognition part of our prototype is very sketchy. It uses simple template matching for phoneme recognition, and heuristic rules to recognize each word. It works reliably for a specific target user (an author) and for limited number of words only. In the future we hope to combine our interaction techniques with a robust speech recognition system and to perform a formal user study.

DISCUSSIONS

The advantages of our approach over traditional speech recognition are 1) immediate, continuous control, 2) language independency, and 3) simplicity. Because voice-as-sound techniques rely on very simple signal processing, it is relatively easy to achieve robust responses in noisy environments.

Our techniques are useful in situations where the user cannot use his or her hands for controlling applications because of permanent physical disability or temporal task-induced disability. Examples include controlling a navigation system while driving a car [8], controlling applications in immersive environments while both hands are doing something else, and controlling wearable or portable devices in outdoor situations. Another promising application area is entertainment. We have implemented simple video games using voice-as-sound techniques, and people found it quite engaging. We also plan to apply our techniques to audience interaction (volume may be more suitable than pitch in this case).

The limitation of this technique is that it requires an unnatural way of using the voice. While traditional speech-based interfaces try to mimic natural human-to-human conversation, voice-as-sound techniques require an artificial way of making vocal sound. Continuously making vocal sound also tires the throat. We found that breathed sound is less straining for long-term interaction and less annoying for other people.

Voice-as-sound techniques *complement* traditional speech recognition interfaces rather than replacing them, by allowing the user to directly adjust system parameters. We continue to explore various combinations of these two approaches to achieve efficient interaction.

REFERENCES

1. Goto M., Itou,K., Akiba,T., Hayamizu,S. Speech Completion: New Speech Interface with On-demand Completion Assistance, Proc. of HCI International 2001, 2001.(in press)
2. Hirose,Y., Ozeki,K., Takagi,K., Effectiveness of prosodic features in syntactic analysis of read Japanese sentences, Proceedings of ICSLP2000, Vol.3, pp.215-218, 2000.
3. Igarashi,T., Hinckley,K. Speed-dependent automatic zooming for browsing large documents, Proceedings of UIST'00, pp.139-148, 2000.
4. Iwano,K., Hirose,K., Prosodic Word Boundary Detection Using Statistical Modeling of Moraic Fundamental Frequency Contours and Its Use for Continuous Speech Recognition, Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol.1, pp.133-136, 1999.
5. Lieske,C., Bos,J., Emele,M., Gamback,B., Rupp,CJ., Giving prosody a meaning, Eurospeech97 vol3 pp.1431-1434, 1997.
6. Manaris,B., McCauley,R., MacGyvers,V., An Intelligent Interface for Keyboard and Mouse Control--Providing Full Access to PC Functionality via Speech, Proceedings of 14th International Florida AI Research Symposium (FLAIRS-01), 2001, (to appear).
7. Tsukahara,W., Ward,N., Responding to Subtle, Fleeting Changes in the User's Internal State. Proceedings of CHI 2001, pp.77-84, 2001.
8. Westphal,M., Waibel,A. Towards Spontaneous Speech Recognition For On-Board Car Navigation And Information Systems, Proceedings of the Eurospeech 99, 1999.