The Vocal Joystick Demo at UIST05: A Voice-Based Human-Computer Interface *

Jeff A. Bilmes, Xiao Li, Jonathan Malkin, Kelley Kilanski, Richard Wright, Katrin Kirchhoff, Amarnag Subramanya, Susumu Harada, James A. Landay, Patricia Dowden, Howard Chizeck
Department of Electrical Engineering Department of Linguistics
Department of Computer Science Department of Speech & Hearing Science
University of Washington,
Seattle, WA
bilmes@ee.washington.edu

We will demonstrate a novel voice-based human-computer interface we call the Vocal Joystick (VJ), designed to enable individuals with motor impairments to use vocal parameters for both discrete and continuous control tasks. Since discrete spoken commands are not ideally suited to such tasks, our methodology exploits a large set of continuous acoustic-phonetic parameters like pitch, loudness, vowel quality, etc. Their selection is optimized with respect to automatic recognizability, communication bandwidth, learnability, suitability, and ease of use. These parameters are extracted continually in real time, transformed via adaptation and acceleration, and converted into continuous control signals. Our UIST'2005 demo will allow the user to try out the vocal joystick on arbitrary web-browsing tasks, VJ-enabled games, and other VJ-based applications.

1 THE VOCAL JOYSTICK

The VJ approach has three main characteristics: **1)** Continuous control parameters: Unlike standard speech recognition, the VJ engine exploits continuous vocal characteristics that go beyond the capabilities of sequences of discrete speech sounds and include e.g. pitch, vowel quality, and loudness, which are mapped to continuous control parameters. **2)** Spoken language: Unlike natural speech, the VJ input "language" is based on a designed set of sounds. These sounds are selected with respect to acoustic discriminability (maximizing accuracy), pronounceability (to reduce potential vocal strain), mnemonic characteristics (to reduce cognitive load), robustness to environmental noise, and based on suitability to the currently running application (e.g., a VJ-based mouse might use only three discrete sounds corresponding to mouse clicks). **3)** Reusable infrastructure: Our goal is not to create a single application but to provide a modular library that can be incorporated into any application requiring vocal control. VJ technology is not meant to replace but rather to complement standard speech recognition technology.

*This material is based upon work supported by the National Science Foundation under Grant No. IIS-0326382

2 The VJ Engine

We have developed a portable modular library (the VJ engine) that can be incorporated into a variety of applications such as mouse and menu control, or robotic arm manipulation. Our design goal is to be modular, low-latency, and as computationally efficient as possible. For example, we share common signal processing operations in multiple signal extraction modules, which yields real-time performance but leaves considerable computational headroom for the applications being driven by the VJ engine.

2.1 Vocal Characteristics

In addition to *discrete sound classes* (e.g., syllables), four vocal characteristics are currently extracted by the VJ engine: *energy*, *pitch*, *vowel quality*, yielding four simultaneously and independently specifiable degrees of freedom. The first of these, localized acoustic energy, is used for voice activity detection. In addition, it is normalized relative to the current vowel detected, and is used by our mouse application to control cursor velocity. For example, a loud voice causes a large movement while a quiet voice causes a "nudge." The second parameter, pitch, is also extracted but is currently unused in existing applications (and thus constitutes a "free parameter" available for future VJ applications). The third parameter is vowel quality. Unlike consonants, which are characterized by a greater degree of constriction in the vocal tract and which are inherently discrete in nature, vowels are highly energetic and therefore are well suited for environments where both high accuracy and noise-robustness is crucial. We map vowels onto the 2-D vowel space characterized by tongue height and tongue advancement (Figure 1). In our VJ-mouse system, we use the four corners of this chart to map to the 4 principle directions of up, down, left, and right as shown in Figure 2.

Finally, discrete sounds are selected according to both linguistic and system criteria. We incorporate a rejection mechanism of sounds corresponding to none of the known discrete sounds. This is indispensable since the input may easily include extraneous speech, non-speech

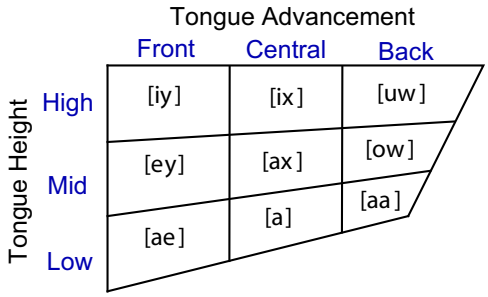

Figure 1: Vowel configurations as a function of their dominant articulatory configurations.

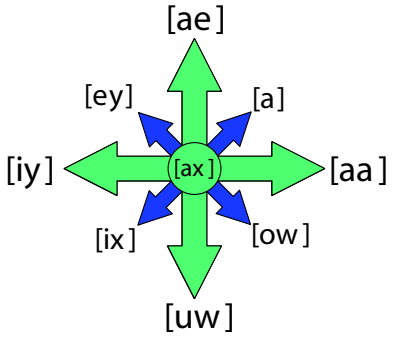

Figure 2: Vowel-direction mapping: vowels corresponding to directions for mouse movement in the WIMP VJ cursor control. The four major direction vowels are: [ae] ("c*a*t") for vertical up movement; [uw] ("b*oo*t") for vertical down movement; [aa] ("h*o*d") for right movement; and [iy] ("h*ee*d") for left movement. The additional four minor (or diagonal) direction vowels (chosen to be roughly a combination of the major vowels) and used for the 8- (or 9-) category classifier are: [a] (German pronunciation of "k*a*trin") for right-up movement; [ow] ("h*oe*d") for right-down movement; [ey] ("h*ay*ed") for left-up movement; and [ix] ("tul*i*p") for left-down movement. Also, we use a schwa [ax] ("*a*go") for a central non-movement vowel in the 9 (or 5) category classifier, used as a carrier for when other parameters (pitch and/or amplitude) are to be controlled without any positional change.

(e.g., breath), and other noise.

3 Demo, Script, and Outline

It is difficult to provide a precise script that users will follow when using the VJ system for our UIST'05 demo. Rather, what we will do is allow the user to do arbitrary web browsing using *only* the voice via the VJ system as control, and therefore the user of the system can browse any desired web page. We will thus need fast internet access for performing this demonstration.

Please see the enclosed video to get a better idea of the VJ system. The video shows four different examples: web browsing (using a New York Times page that is link heavy, so voice tabbing is inefficient); a voice-controlled video game; browsing using google maps; and a VJ-based voice visualization tool. A number of other videos are available at the following: `http://ssli.ee.washington.edu/vj/video_demos.htm`.

For our demo, we will make available all of the above applications, as well as interface the VJ with the Dasher system [4], something we call the "Vocal Dasher." We have also interfaced the VJ system directly into a simple blocks world environment, where more precise object movement and cursor position measurement is possible than via the mouse driver. This will also be available.

3.1 Comparisons to Related Work

There are a number of systems that have used the human voice in novel ways for controlling mouse movement. We point out, however, that the VJ is conceptually different than the other systems in several important respects, and this includes both *latency* and *design*. First, VJ overcomes the *latency* problem in vocal control. VJ allows the user to make instantaneous directional changes using one's voice (e.g., the user can dynamically draw a "U" or "L" shape in one breath). Olwal and Feiner's system [3] moves the mouse only after recognizing entire words. In Igarashi's system [2], one needs first to specify direction, and then afterwards a sound to move in the said direction. De Mauro's system [1] moves the mouse after the user has finished vocalizing. The VJ, by contrast, has latency (time between control parameter change in response to a vocal change) on the order of reaction time (currently, approximately 60 ms), so direction and other parameters can change during vocalization. The other key difference from previous work is that VJ is general software infrastructure, *designed* from the outset not only for mouse control, but also for controlling a robotic arm, wheelchair, normal joystick signals, etc. A VJ system is customizable, e.g., the vowel-to-space mapping can be changed by the user. Our software system, moreover, is generic. It outputs simultaneous control parameters corresponding to vowel quality, pitch, formants (F1/F2), and amplitude (i.e., we even have unused degrees of freedom in the mouse application). The system can be plugged into either a mouse driver or any other system.

4 REFERENCES

1. C. de Mauro, M. Gori, M. Maggini, and E. Martinelli. A voice device with an application-adapted protocol for microsoft windows. In *Proc. IEEE Int. Conf. on Multimedia Comp. and Systems*, volume II, pages 1015—1016, Firenze, Italy, 1999.

2. T. Igarashi and J. F. Hughes. Voice as sound: Using non-verbal voice input for interactive control. In *ACM UIST 2001*, November 2001.

3. Alex Olwal and Steven Feiner. Interaction techniques using prosodic features of speech and audio localization. In *IUI '05: Proc. 10th Int. Conf. on Intelligent User Interfaces*, pages 284–286, New York, NY, USA, 2005. ACM Press.

4. D.J. Ward, A. F. Blackwell, and D.J. C. MacKay. Dasher - a data entry interface using continuous gestures and language models. In *ACM UIST 2000*, 2000.