THE VOCAL JOYSTICK

Jeff A. Bilmes[†], Jonathan Malkin[†], Xiao Li[†], Susumu Harada[§], Kelley Kilanski[‡],
Katrin Kirchhoff[†], Richard Wright[‡], Amarnag Subramanya[†], James A. Landay[§],
Patricia Dowden[¶], Howard Chizeck[†]

[†]Dept. of Electrical Engineering [‡]Dept. of Linguistics
[§]Dept. of Computer Science & Eng. [¶]Dept. of Speech & Hearing Science
University of Washington
Seattle, WA

ABSTRACT

The *Vocal Joystick* is a novel human-computer interface mechanism designed to enable individuals with motor impairments to make use of vocal parameters to control objects on a computer screen (buttons, sliders, etc.) and ultimately electro-mechanical instruments (e.g., robotic arms, wireless home automation devices). We have developed a working prototype of our “VJ-engine” with which individuals can now control computer mouse movement with their voice. The core engine is currently optimized according to a number of criterion. In this paper, we describe the engine system design, engine optimization, and user-interface improvements, and outline some of the signal processing and pattern recognition modules that were successful. Lastly, we present new results comparing the vocal joystick with a state-of-the-art eye tracking pointing device, and show that not only is the Vocal Joystick already competitive, for some tasks it appears to be an improvement.

1. INTRODUCTION

Many existing human-computer interfaces (e.g., the mouse and keyboard, touch screens, pen tablets, etc.) are ill-suited to individuals with motor impairments. Specialized (and often expensive) human-computer interfaces have been developed specifically for this group, including sip-and-puff switches [1], head mice [2, 1, 3], eye-gaze and eye tracking devices [4], chin joysticks [5], and tongue switches [6]. While many individuals with motor impairments have complete use of their vocal system, these assistive devices do not make full use of it. Sip and puff switches, for example, control a device by sending binary signals and thus have relatively low communication bandwidth, making it difficult to perform complex control tasks.

Natural spoken language is regarded as an obvious choice for a human-computer interface. Despite significant research efforts in automatic speech recognition (ASR), however, existing ASR systems are still not perfectly robust to a wide variety of speaking conditions, noise, and accented speakers, and they have not yet been universally adopted as a dominant human-computer interface. In addition, while natural speech is optimal for human-to-human communication, it may be sub-optimal for manipulating computers, windows-icons-mouse-pointer (WIMP) interfaces, and other electro-mechanical devices (such as a prosthetic robotic arm). Standard spoken language commands, moreover, are ideal for discrete but not for continuous operations. For example, to move a cursor from the bottom-left to the upper-right of a screen, a user could repeatedly utter “up” and “right”, or alternatively “stop” and “go” after setting an initial trajectory and rate, but this can be inefficient. Other methods for using controlling mouse movement with speech have also been developed [7, 8, 9] but none of these take advantage of the full continuous nature of the human vocal system.

For the above reasons, we have developed an alternative voice-based assistive technology termed the *Vocal Joystick* (VJ) [10]. Unlike standard ASR, our system goes beyond the capabilities of sequences of discrete speech sounds, and exploits continuous vocal characteristics such as pitch, vowel quality, and loudness which are then mapped to continuous control parameters. Several video demonstrations of the Vocal Joystick system are available online `http://ssli.ee.washington.edu/vj`. In previous work, we gave a high-level overview of the vocal joystick [10] and details regarding motion acceleration [11] and adaptation [12, 13]. In this work, we provide details about our design goals, the signal processing and pattern recognition modules, and we report on a new user-study that shows that the VJ compares favorably to a standard modern eye tracking device.

2. VOCAL JOYSTICK

The Vocal Joystick system maps from human vocalic effort to a set of control signals used to drive a mouse pointer or robotic arm. It also allows a small set of discrete spoken commands usable as mouse clicks, button presses, and modality shifts. We use a “joystick” as an analogy since it has the ability to simultaneously specify several continuous degrees of freedom along with a small number of button presses, and we consider this to be a generalization of a mouse.

In developing the VJ system, we have drawn on our ASR background to produce a system that, as best as possible, meets the following goals: **1) easy to learn:** the VJ system should be easy to learn and remember in order to keep cognitive load at a minimum. **2) easy to speak:** using a VJ-controlled device should not produce undue strain on the human vocal system. It should be possible to use the system for many hours at a time. **3) easy to recognize:** the VJ system should be as noise robust as possible, and should try to include vocal sounds that are as acoustically distinct as possible. **4) perceptual:** the VJ system should respect any perceptual expectations that a user might have and also should be perceptually consistent (e.g., given knowledge of some aspects of a VJ system, a new vocal effort should, say, move the mouse in an expected way). **5) exhaustive:** to improve communications bandwidth, the system should utilize as many capabilities of the human vocal apparatus as possible, without conflicting with goal 1. **6) universal:** our design should use vocal characteristics that minimize the chance that regional dialects, or ac-

This material is based on work supported by the National Science Foundation under grant IIS-0326382.

cents will preclude its use. **7) complementary:** the system should be complementary with existing ASR systems. We do not mean to replace ASR, but rather augment it. **8) resource-light:** a VJ system should run using few computational resources (CPU and memory) and leave sufficient computational headroom for a base application (e.g., a web browser, spreadsheet). **9) infrastructure:** the VJ system should be like a library, that any application can link to and use.

Unlike standard speech recognition, the VJ engine exploits the ability of the human voice to produce continuous signals, thus going beyond the capabilities of sequences of discrete speech sounds (such as syllables or words). Examples of these vocal parameters include pitch variation, type and degree of vowel quality, and loudness. Other possible (but not yet employed) qualities are degree of vibrato, low-frequency articulator modulation, nasality, and velocity and acceleration of the above.

2.1. Primary Vocal Characteristics

Three continuous vocal characteristics are currently extracted by the VJ engine: *energy*, *pitch*, and *vowel quality*, yielding four simultaneous 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 detected vowel, and is used by our mouse application to control the velocity of cursor movement. 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 (it thus constitutes a free parameter available for future use). 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 thus are well suited for environments where both high accuracy and noise-robustness are crucial. Vowels can be characterized using a 2-D space parameterized by F1 and F2, the first and second vocal tract formants (resonant frequencies). We classify vowels, however, directly and map them onto the 2-D vowel space characterized by tongue height and tongue advancement (Figure 1) (we found F1/F2 estimation to be too unreliable for this application). In our initial VJ system, and in our VJ mouse control, 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 1. We have also produced an 8- and 9-class vowel system to enable more non-simultaneous degrees of freedom and more precise specification of diagonal directions. We also utilize a “neutral” schwa [ax] as a carrier vowel for when other parameters (pitch and/or amplitude) are to be controlled without any positional change. These other vowels (and their directional correlates) are also shown in Figure 1.

In addition to the three continuous vocal parameters, “discrete sounds” are also employed. We select (or reject) a candidate discrete sound according to both linguistic criteria and the system criteria mentioned in Section 2. So far, however, we have not utilized more than two or three discrete sounds since our primary application has been mouse control. Our research has thus focused on real-time extraction of continuous parameters since that is less like standard ASR technology.

3. THE VJ ENGINE

We have developed a portable modular library (the VJ engine) that can be incorporated into a variety of applications. Following the goals from Section 2, the engine shares common signal processing operations in multiple modules, to produce real-time performance while leaving considerable headroom for the applications being driven by the VJ engine.

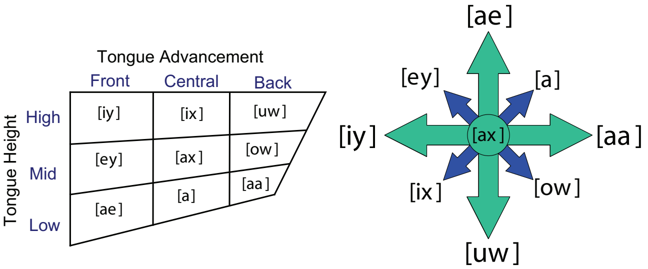

Fig. 1. Left:Vowel configurations as a function of their dominant articulatory configurations. Right: Vowel-direction mapping: vowels corresponding to directions for mouse movement in the WIMP VJ cursor control.

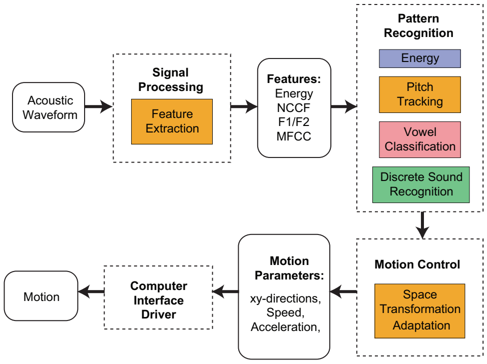

Fig. 2. The vocal joystick engine system structure.

The VJ engine consists of three main components: *acoustic signal processing*, *pattern recognition*, and *motion control* (see Figure 2). First, the signal processing module extracts short-term acoustic features, such as energy, autocorrelation coefficients, linear prediction coefficients and mel frequency cepstral coefficients (MFCCs). These features are piped into the pattern recognition module, where energy smoothing, pitch and formant tracking, vowel classification and discrete sound recognition take place. This stage also involves pattern recognition methods such as neural networks, support vector machines (SVMs), and dynamic Bayesian networks (see [12, 14, 15]). Finally, energy, pitch, vowel quality, and discrete sounds become acoustic parameters that are transformed into direction, speed, and other motion-related parameters for the back-end application.

An important first stage in the signal processing module is *voice activity detection* (VAD). We categorize each frame into the three separate categories *silence*, *pre-active*, or *active*, based on energy and zero-crossing information. Pre-active frames may (or may not) indicate the beginning of voice activity, for which only frontend feature extraction is executed. Active frames are those identified as truly containing voice activity. Pattern recognition tasks, including pitch tracking and vowel classification, are performed for these frames. No additional computation is used for silence frames. If silence frames occur after an unvoiced segment within a length range, however, discrete sound recognition will be triggered.

The goal of the signal processing module is to extract *low-level acoustic features* that can be used in estimating the four high-level acoustic parameters. The acoustic waveforms are sampled at a rate of 16,000 Hz, and a frame is generated every 10 ms. The extracted frame-level features are energy, normalized cross-correlation coefficients (NCCC), formants, and MFCCs [16, 17]. In addition, we employ delta features and online (causal) mean subtraction and variance normalization.

Our *pitch tracker* module is based on several novel ideas. Many pitch trackers require meticulous design of local and transition costs. The forms of these functions are often empirically determined and their parameters are tuned accordingly. In the VJ project, we use a graphical modeling framework to automatically optimize pitch tracking parameters in the maximum likelihood sense. Specifically, we use a dynamic Bayesian network to represent the pitch- and formant-tracking process and learn the costs using an EM algorithm [14, 15]. Experiments show that this framework not only expedites pitch tracker design, but also yields good performance for both pitch/F1/F2 estimation and voicing decision.

Vowel classification accuracy is crucial for overall VJ performance since these categories determine motion direction. Additional requirements include real-time, consistent, and noise robust performance. Vowel classification in the VJ framework differs from conventional phonetic recognition in two ways: First, the vowels are longer duration than in normal speech. Second, instantaneous classification is essential for real-time performance. In our system, we utilize posterior probabilities of a discriminatively trained multi-layer perceptron (MLP) using MFCC features as input. We have also developed a novel algorithm for real-time adaptation of the MLP and SVM parameters [12, 13], this increases the accuracy of our VJ classifier considerably!

We have also found that *acceleration* has yielded significant improvements to VJ performance [11]. Unlike normal mouse acceleration, which adjusts a mapping from a 2-D desktop location to a 2-D computer screen location, a VJ system must map from vocal tract articulatory change to positional screen changes. We utilize the idea of “intentional loudness,” where we normalize energy based on how a user intends to affect the mouse pointer, and have developed a non-linear mapping that has shown in user studies to be preferable to no vocal acceleration.

Our *discrete sound recognition* module uses a fairly standard HMM system. We currently use consonant-only patterns for discrete sounds, like /ch/ and /t/ (sufficient for a 1-button mouse), and we use a temporal threshold to reject extraneous speech. This not only significantly reduces false positives (clicks), but also saves computation since only pure unvoiced segments of a certain length will trigger the discrete sound recognition module to start decoding.

3.1. User Study: Comparison of VJ and Eye Tracker

We performed a study comparing a VJ-mouse with a standard eye tracking mouse. Specifically, we investigated the difference in users' performance between the Vocal Joystick (VJ) system and the eye tracker (ET) system.

The eye tracker consisted of a single infrared camera that was designed to focus on the users dominant eye and to track the movement of the iris by analyzing the reflection of an infrared beam emanating from the camera. This particular system required the user's head to stay fairly steady, so we utilized a chin rest for the participants to rest their chin and reduce the amount of head movement (something not needed by a VJ-based system). Clicking is performed by dwelling, or staring at the desired point for a fixed amount of time. The dwelling time threshold is configurable, but we used the default (0.25 seconds) throughout the experiment.

We recruited 12 participants from the UW Computer Science department to participate in our experiment. Of the 12, there were five females and seven males, ranging in age from 21 to 27. Seven of the participants were native English speakers and the rest of them were from Europe and Asia. Five participants wore glasses, two wore contact lenses, and the rest had uncorrected vision. We exposed each participant to two different modalities: VJ and ET. For each modality, we had the participants perform two tasks: Target Acquisition task (TA) and the Web Browsing task (WB). The order of the tasks within each modality was fixed (TA then WB). The participants completed both tasks under one modality before moving on to the other modality. Before starting on any task for each modality, the participants were given a description of the system they were about to use, followed by a calibration phase (for VJ, the participants were asked to vocalize the four vowels for two seconds each; for ET, the participants were asked to look at a sequence of points on the screen based on the Eye Tracker calibration software). They were then given 90 seconds to try out the system on their own to get familiar with the controls.

All experimental conditions were shown on a 19-inch 1024x768 24-bit color LCD display. The VJ system was running on a Dell Inspiron 9100 laptop with a 3.2 GHz Intel Pentium IV processor running the Fedora Core 2 operating system. A head-mounted Andrea NC-61 microphone was used as the audio input device. The ET system was running on a HP xw4000 desktop with a 2.4 GHz Intel Pentium IV processor running Windows XP Service Pack 2. The ET camera was Eye Response Technologies WAT-902HS model, and the software was Eye Response Technologies ERICA version

For the TA tasks, we wrote an application that sequentially displays the starting point and the target for each trial within a maximized window and tracks the users clicks and mouse movements. A Firefox browser was used for the WB tasks. The browser was screen maximized such that the only portion of the screen not displaying the contents of the web page was the top navigation toolbar, which was 30 pixels high.

The TA task consisted of sixteen different experimental conditions with one trial each. A trial consisted of, starting at a fixed region at the center of the screen (a 30 pixel wide square), attempting to click on a circular target which appears with a randomized size, distance, and angle from the center region.

The WB task consisted of one trial in which the user was shown a sequence of web links that they needed to click through and was told to follow the same links using a particular modality (VJ or ET). The participants were first guided through the sequence by the experimenter, and then asked to go through the links themselves to ensure that they were familiar with the order and the location of each link. They were also instructed that if they click on a wrong link, they must click on the browsers back button on their own to return to the previous page and try again. Once the participant was familiar with the link sequence, they were asked to navigate through those links using a particular modality. The time between when the participant started using the modality and when the participant successfully clicked on the last link was recorded, as well as the number of times they clicked on a wrong link. The sequence of links consisted of clicking on six links starting from the CNN homepage `www.cnn.com`. Most of the links were 15 pixels high and link widths ranged from 30 to 100 pixels wide, and distances between links ranged from 45 pixels to 400 pixels, covering directions corresponding roughly to six different approach angles.

The mean task completion speed (inverse time) across all participants for each of the 16 conditions across the two modalities is shown in Figure 3. The higher bars on the graphs indicate faster performance. The circles represent the target size, distance, and angle relative to the start position (middle square) for all conditions. The error bars represent the 95^{th} percentile confidence interval (as with the other error bars in the other figure in this section). The mean

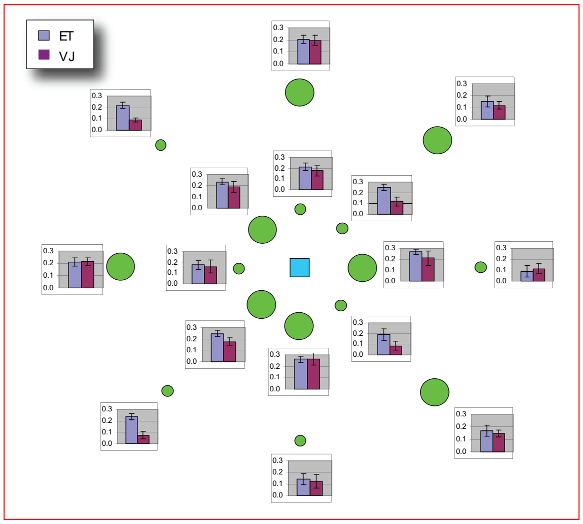

Fig. 3. Mean task completion “speed” (1/seconds) for the TA task across modalities.

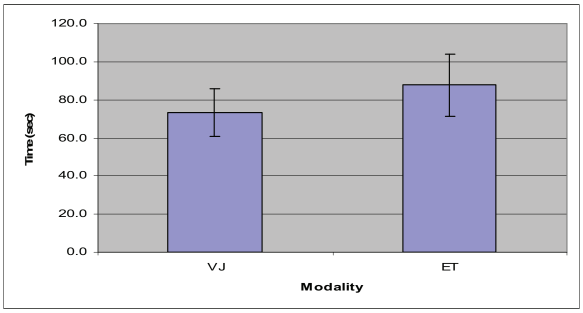

Fig. 4. Web browsing task completion times (sec) across modalities

task completion times (which include any missed-link error recovery time) for the web browsing task are shown in Figure 4.

Overall, our results suggest that the Vocal Joystick allowed the users to perform simple TA tasks at a comparable speed as the particular eye tracker we used, and that for the web browsing task, the VJ was significantly faster than ET. This is quite encouraging given that the VJ system is quite new!

3.2. 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 Vocal Joystick 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 [8] moves the mouse only after recognizing entire words. In Igarashi’s system [7], one needs first to specify direction, and then afterwards a sound to move in the said direction. De Mauro’s system [18] 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 robotic arms, wheelchairs, 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 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] “Pride mobility products group sip-n-puff system/head array control,” 2005, http://www.pridemobility.com.

[2] “Origin instruments sip/puff switch and head mouse,” 2005, orin.com/access/headmouse/index.htm.

[3] “Headmaster head mouse,” 2003, http://www.wati.org/headmaster.htm.

[4] “Assistive technologies’s eye gaze system for computer access,” 2003, http://www.assistivetechnologies.com/proddetails/EG001B.htm.

[5] “Chin joystick,” 2003, http://www.pulsar.org/archive/eyal/fal97/fal97_headset.html.

[6] “Prentrom devices,” 2003, http://store.prentrom.com/catalog/prentrom/caswitches.html.

[7] T. Igarashi and J. F. Hughes, “Voice as sound: Using non-verbal voice input for interactive control,” in *14th Annual Symposium on User Interface Software and Technology*. ACM UIST’01, November 2001.

[8] A. Olwal and S. Feiner, “Interaction techniques using prosodic features of speech and audio localization,” in *IUI ’05: Proc. 10th Int. Conf. on Intelligent User Interfaces*. New York, NY, USA: ACM Press, 2005, pp. 284–286.

[9] “Dragon naturally speaking, Mousegrid^{TM}, ScanSoft Inc.” 2004.

[10] J. Bilmes, X. Li, J. Malkin, K. Kilanski, R. Wright, K. Kirchhoff, A. Subramanya, S. Harada, J. Landay, P. Dowden, and H. Chizeck, “The vocal joystick: A voice-based human-computer interface for individuals with motor impairments,” in *Human Language Technology Conf. and Conf. on Empirical Methods in Natural Language Processing*, Vancouver, October 2005.

[11] J. Malkin, X. Li, and J. Bilmes, “Energy and loudness for speed control in the vocal joystick,” in *Proc. IEEE Automatic Speech Recognition and Understanding (ASRU)*, Nov. 2005.

[12] X.Li, J.Bilmes, and J.Malkin, “Maximum margin learning and adaptation of MLP classifers,” in *9th European Conference on Speech Communication and Technology (Eurospeech’05)*, Lisbon, Portugal, September 2005.

[13] X.Li and J.Bilmes, “Regularized adaptation of discriminative classifiers,” in *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, 2006.

[14] X.Li, J.Malkin, and J.Bilmes, “A graphical model approach to pitch tracking,” in *Proc. Int. Conf. on Spoken Language Processing*, 2004.

[15] J. Malkin, X. Li, and J. Bilmes, “A graphical model for formant tracking,” in *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, 2005.

[16] J. Deller, J. Proakis, and J. Hansen, *Discrete-time Processing of Speech Signals*. MacMillan, 1993.

[17] X. Huang, A. Acero, and H.-W. Hon, *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. Prentice Hall, 2001.

[18] 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*, vol. II, Firenze, Italy, 1999, pp. 1015—1016.