Speech Shift: Direct Speech-Input-Mode Switching through Intentional Control of Voice Pitch

Masataka Goto†, Yukihiro Omoto††, Katunobu Itou†, and Tetsunori Kobayashi††

† National Institute of Advanced Industrial Science and Technology (AIST).
1-1-1 Umezono, Tsukuba, Ibaraki 305-8568, JAPAN.

{m.goto@, itou@ni.}aist.go.jp

†† Dept. EECE, Waseda University. Shinjuku-ku, Tokyo 169-8555, JAPAN.

{omoto, koba}@tk.elec.waseda.ac.jp

Abstract

This paper describes a speech-input interface function, called *speech shift*, that enables a user to specify a speech-input mode by simply changing (shifting) voice pitch. While current speech-input interfaces have used only verbal information, we aimed at building a more user-friendly speech interface by making use of nonverbal information, the voice pitch. By intentionally controlling the pitch, a user can enter the same word with it having different meanings (functions) without explicitly changing the speech-input mode. Our speech-shift function implemented on a voice-enabled word processor, for example, can distinguish an utterance with a high pitch from one with a normal (low) pitch, and regard the former as *voice-command-mode input* (such as file-menu and edit-menu commands) and the latter as *regular dictation-mode text input*. Our experimental results from twenty subjects showed that the speech-shift function is effective, easy to use, and a labor-saving input method.

1. Introduction

Current speech-input interfaces have not fully exploited the potential of speech. Although human speech has two aspects, verbal information (e.g., words) and nonverbal information (e.g., pitch and hesitation), most speech recognizers utilize only the verbal information of speech input. The purpose of this study is to build a user-friendly speech interface that makes full use of nonverbal speech information intentionally controlled by a user.

While nonverbal speech information plays valuable roles in human-human communication, its use for speech recognition or understanding has been limited. Several papers have reported that prosodic information raises the rank of the correct hypothesis in speech recognition [1], reduces the word error rates of a large-vocabulary speech recognizer [2], and increases Japanese *mora* (a unit similar to a syllable) recognition rates [3]. Also, the Verbmobil project [4] has succeeded in using prosodic information, mainly for the analysis of phrase boundaries. These studies, however, dealt with auxiliary aspects of nonverbal information unintentionally contained in natural speech input, and did not use nonverbal information for interface functions.

In this paper we describe a mode-switching function for speech input, called *speech shift*, which enables a user to switch speech-input modes by intentionally changing the pitch of an utterance. The speech-shift function allocates two types of utterances — an utterance with a normal (low) pitch and one with a high pitch — to different speech-input modes, such as the dictation and voice-command modes on a voice-enabled word processor. This function can enable seamless speech-input-mode switching without changing the mode explicitly, while most current speech-input interfaces require explicit mode switching by using key phrases or other devices. The most important point is

that we use *intentional* pitch control for the speech-shift function: this brings additional information to speech-input interfaces.

In the following sections, we explain the basic concept of speech shift and then describe the design and implementation of a voice-enabled word processor with the speech-shift function. Finally, we show that experimental results from twenty subjects indicated the effectiveness of speech shift.

2. Speech shift

Speech shift is a speech-interface function that enables a user to directly enter a word in the intended speech-input mode even when the word can be accepted in different modes on a speech interface. Current speech-input interfaces cannot distinguish a word from the same one with a different pitch because they recognize only verbal (phoneme) information. The speech-shift function can distinguish between them and allocate them to different speech-input modes.

On a voice-enabled word processor system, for example, a problem arises when the system is given an unaccompanied user utterance "save" — the system cannot judge whether the user wants to enter the text of "save" or execute the voice command of "save." A typical approach to solving this problem is to prepare two speech-input modes — the dictation mode for continuous speech dictation and the voice-command mode for executing file-menu and edit-menu commands — and explicitly switch between them by using predefined key phrases, such as "dictation" and "voice command," or by using other devices such as a mouse or keyboard.

The speech-shift function solves this problem by regarding an utterance with a normal (low) pitch (called a *normal utterance*) as regular dictation-mode text input and regarding an utterance with a high (shifted) pitch (called a *shift utterance*) as voice-command-mode input. By changing the pitch of the utterance "save", a user can tell the system whether it should be accepted in the dictation or voice-command mode. The speech-shift function provides two benefits:

1. Switching without other devices

   A user can invoke functions in different speech-input modes without needing to use other devices to switch between those modes.

2. Seamless switching between speech-input modes

   A user can seamlessly invoke functions in different speech-input modes without switching between those modes explicitly and without needing to be aware of the current speech-input mode.

3. Method of distinguishing between normal and shift utterances

The speech-shift function requires that a distinction be made between normal (low-pitch) utterances and shift (high-pitch) utterances. It is, however, difficult to judge whether the pitch of an utterance is intentionally shifted (raised) because the pitch range of voices differs among individuals.

We therefore introduce a unique pitch reference for each speaker, called the *base fundamental frequency (base F0)*, which represents the pitch of the speaker's natural voice. After estimating the base F0 for each speaker, we can deal with the pitch value relative to the base F0 instead of the absolute value of the voice pitch. If the *relative pitch value* of an utterance, which is calculated by subtracting the base F0 from the average pitch of the utterance, is high enough, the pitch of the utterance is judged to be intentionally shifted.

3.1. Method of estimating the base F0

We propose a method of estimating the base F0 by averaging the voice pitch during a filled pause such as "er..." or "uh..." (the lengthening of a vowel during hesitation). Since the filled pause is a natural hesitation that indicates a speaker is having trouble preparing (thinking of) a subsequent utterance, the speaker cannot change articulator parameters (the positions and states of the articulators, including the larynx) during filled pauses [5]. We can therefore assume that the pitch during filled pauses is stable and is very close to the pitch of the speaker's natural voice, i.e., the base F0. In addition, because filled pauses often occur during natural speech input, this approach of calibrating the base F0 has an advantage that a speaker can utter filled pauses easily and effortlessly and that the method can gradually update the base F0 for every filled pause. This update is achieved by using the maximum *a posteriori* probability (MAP) estimation.

To detect filled pauses and estimate the voice pitch in real time, we use a robust filled-pause detection method [5]. This is a bottom-up method that can detect a lengthened vowel in any word without using top-down information (a language model). It determines the beginning and end of each filled pause by finding two acoustical features of filled pauses — small fundamental frequency transitions and small spectral envelope deformations. These features are found by estimating the voice pitch (the fundamental frequency) through a sophisticated instantaneous-frequency-based analysis [5]: we find the most predominant harmonic structure in extracted frequency components by using a comb-filter-like analysis.

3.2. Analysis of F0 during filled pauses

As a preliminary experiment, we examined the stability of the fundamental frequency (F0) (i.e., the voice pitch) during filled pauses. Figure 1 shows the average F0 for three typical Japanese fillers — /n-/, /e-/, and /ano-/ — uttered by six Japanese male subjects. The standard deviation of the F0 during filled pauses was 86.2 [cent][1] on average. These results show that the average F0 differed among speakers but was stable for each speaker even if different fillers were uttered. We therefore concluded that it is necessary to estimate the base F0 for each speaker and that the average F0 during filled pauses is stable enough to be used for estimating the base F0.

3.3. Speech-shift method using a threshold relative to the base F0

Figure 2 shows an overview of the speech-shift method. The method involves three steps.

---

[1]Frequency f_{Hz} in hertz is converted to frequency f_{cent} in cents so that there are 100 cents to a tempered semitone and 1200 to an octave: $f_{cent} = 1200 \log_2(f_{Hz} / (440 \times 2^{\frac{3}{12}-5}))$.
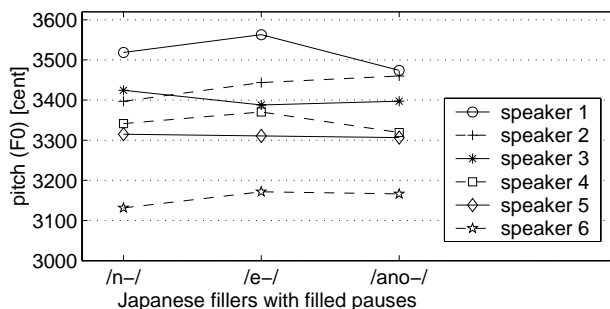

Figure 1: Average F0 for three typical Japanese fillers (with filled pauses) uttered by six speakers.
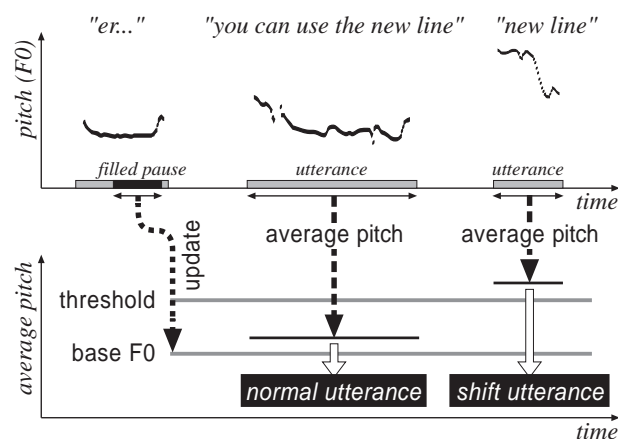

Figure 2: Overview of the speech-shift method using a threshold relative to the base F0.

1. Estimating and updating the base F0
   The base F0 is updated whenever a filled pause is detected.
2. Calculating the relative pitch value for each utterance
   The relative pitch value is obtained by subtracting the base F0 from the pitch averaged over each utterance.
3. Judging whether an utterance is normal or shifted
   The method uses a threshold of the relative pitch value to distinguish between normal and shift utterances. If the relative pitch value is higher than the threshold, an utterance is judged to be a shift utterance; otherwise, it is judged to be a normal utterance.

The threshold is determined in advance to maximize the classification performance for a learning data set.

4. Voice-enabled word processor with the speech-shift function

While the speech-shift function is a general idea that is useful for any voice-enabled application, this section is focused on describing a voice-enabled word processor system we have developed. The system accepts a normal utterance as dictation text and a shift utterance as a voice command. Excerpts of the supported voice commands are:

- edit-menu and format-menu commands
  delete, backspace, bold, left justify, right justify, center justify, new line (enter), undo, cut, paste, etc.
- file-menu commands
  save, open (file open), print, close document, etc.

Since there is no explicit speech-input-mode switching, a user can enter a document efficiently: if a user says "new line" with a high pitch while dictating, it is immediately accepted as a voice command.

While most voice-enabled word processors do not permit a user to hesitate with filled pauses while dictating, our system encourages a user to do this because filled pauses are necessary to estimate the base F0. Utterances with filled pauses are not accepted as dictation: they are used only to estimate the base F0. A user can thus feel comfortable hesitating naturally when thinking of subsequent utterances.

4.1. Speech-shift method incorporating prior knowledge about linguistic context

To improve the ability to distinguish between normal and shift utterances in this word processor system, we developed a method of incorporating prior knowledge about the linguistic context of voice commands — that is, knowledge about positions in which voice commands are likely to be uttered during dictation. For example, even when the method described in Section 3.3 might misjudge from a slightly higher than normal pitch that a non-voice-command utterance should be taken as a shift utterance, we can expect this to be corrected by our method of using prior knowledge.

By extending the standard speech recognition framework in which a word sequence, $W = \{w_1, w_2, \cdots, w_K\}$ (K is the number of words), is obtained by

$$\{\hat{W}\} = \operatorname*{argmax}_{W} P(W|X), \qquad (1)$$

where $X = \{x_1, x_2, \cdots, x_N\}$ is an acoustic observation sequence (spectrum) and N is the number of observation frames (10 ms), this method obtains

$$\{\hat{W}, \hat{C}\} = \operatorname*{argmax}_{W,C} P(W,C|X,A), \qquad (2)$$

where $C = \{c_1, c_2, \cdots, c_K\}$ is a command-flag sequence and $A = \{a_1, a_2, \cdots, a_N\}$ is a pitch sequence. The command flag c_k is 0 for a normal utterance and 1 for a shift utterance. Under the assumption that W and C are independent of, respectively, A and X, Equation (2) can be developed as

$$\operatorname*{argmax}_{W,C} P(W,C|X,A)$$

$$= \operatorname*{argmax}_{W,C} P(C|W,X,A) \cdot P(W|X,A) \qquad (3)$$

$$\cong \operatorname*{argmax}_{W,C} P(C|W,A) \cdot P(W|X) \qquad (4)$$

$$= \operatorname*{argmax}_{W,C} \frac{P(A|C,W) \cdot P(C|W)}{P(A|W)} \cdot P(W|X) \qquad (5)$$

$$\cong \operatorname*{argmax}_{W,C} \frac{P(A|C) \cdot P(C|W)}{P(A)} \cdot P(W|X) \qquad (6)$$

$$= \operatorname*{argmax}_{W,C} P(A|C) \cdot P(C|W) \cdot P(W|X), \qquad (7)$$

where $P(A|C)$ is the word-pitch model and $P(C|W)$ is the command-flag model. The method finally computes Equation (7) by using the N-best rescoring paradigm: since the term $P(W|X)$ corresponds to results of a standard speech recognizer, the method uses it to obtain the N-best list in the first pass, and then rescores the list by using $P(A|C)$ and $P(C|W)$ as well as $P(W|X)$ in the second pass.

4.2. Word-pitch model

The word-pitch model $P(A|C)$ is computed by assuming
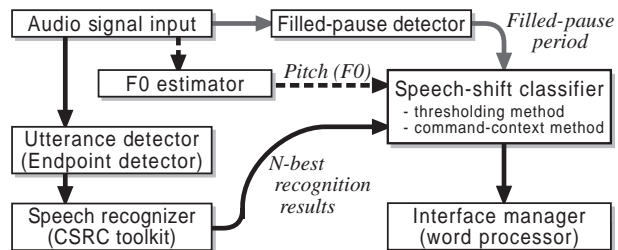
$$P(A|C) \cong \prod_k P(\bar{a}_k|c_k), \qquad (8)$$

Figure 3: System architecture of the speech-shift-enabled word processor.

where $\bar{a}_k$ is the average pitch of a word w_k. $P(\bar{a}_k|c_k)$ represents the distribution of the relative pitch value $\bar{a}_k$ for each category c_k of the normal and shift utterances. This is modeled by a normal (Gaussian) distribution whose parameters are obtained by using maximum-likelihood estimation for a learning data set.

4.3. Command-flag model

By borrowing the idea underlying the class-trigram language model, the command-flag model $P(C|W)$ is computed by assuming

$$P(C|W) \cong \prod_k P(c_k|w_{k-1}, w_k, w_{k+1}) \qquad (9)$$

$$\cong \prod_k P(c_k|v_{k-1}, v_k, v_{k+1}), \qquad (10)$$

where v_k is the word class that w_k belongs to. We defined the following three classes: C (voice-command word), U (non-voice-command word), and S (silent period). This approach reduces the number of learning data sets required to build the command-flag model.

5. Implementation

Figure 3 shows the architecture of our voice-enabled word processor system with the speech-shift function. Each of the seven boxes in the figure represents a different process. These can be distributed over a LAN (Ethernet) and connected by using a network protocol called *RVCP (Remote Voice Control Protocol)*, which is an extension of RMCP [6] that supports timestamp-based synchronization.

The speech recognizer is implemented by using the CSRC (continuous speech recognition consortium) Japanese dictation toolkit [7] which uses the *Julius 3.2* LVCSR (large-vocabulary continuous speech recognizer) engine, a PTM (phonetic tied-mixture) triphone model, and a 20k-word trigram language model. The five-best recognition results are sent to the speech-shift classifier.

The speech-shift classifier receives results from the F0 estimator, filled-pause detector, and speech recognizer, and distinguishes between normal and shift for each utterance by using one of the methods described in either Section 3.3 or Section 4.1. It also updates the base F0 when a filled pause is detected. These results are sent to the interface manager that provides all voice-command functions for the word processor as well as a graphical feedback of the pitch of utterances, the base F0, and the threshold.

6. Experimental results

We evaluated the two proposed methods, the *thresholding method* described in Section 3.3 and the *command-context method* described in Section 4.1, and then evaluated the usability of the voice-enabled word processor system described in Section 4.
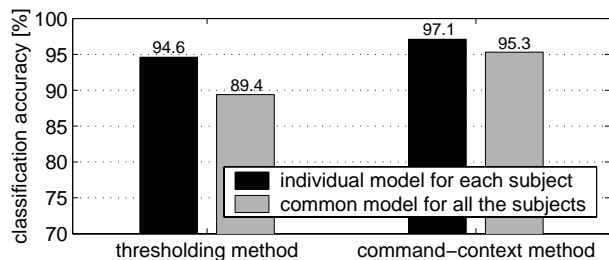
Figure 4: Evaluation results regarding the ability to distinguish between normal and shift utterances.

6.1. Evaluation of speech-shift classification performance

The thresholding and command-context methods were both tested on 120 utterances (60 normal utterances and 60 shift utterances) made by each of twelve Japanese subjects and which included words, phrases, and sentences of various lengths. The classification accuracy was measured with three-fold cross validation. Both methods were tested under two conditions: 1) using an individual model (threshold) optimized for each subject, and 2) using a common model (threshold) optimized for all the subjects. The command-flag model of the command-context method was trained on operational logs of our word processor.

Figure 4 shows the evaluation results regarding the classification accuracy of the two methods. For both methods, the results under the individual-model condition were superior to those under the common-model condition. The better performance of the command-context method compared to that of the thresholding method shows that introducing prior knowledge about the linguistic context is effective when it is available. These results show that both methods are robust enough to distinguish between normal and shift utterances and to make the speech-shift function practical.

6.2. Usability evaluation of the speech-shift-enabled word processor

We tested our word processor system with twenty Japanese subjects who were experienced in using word processors. To evaluate whether the speech-shift function was efficient, we first measured the time each subject required to enter a text document written on a paper sheet under the following four input methods after the subject gained a good command of them: (a) speech-input modes were switched by mouse operation, (b) speech-input modes were switched by uttering predefined key phrases, (c) phrases uttered while the shift key was pressed were accepted as voice commands, and (d) voice commands were directly entered through the speech-shift function (the proposed method). While our system (as described in Section 4) had no explicit speech-input modes, we prepared two speech-input modes — dictation and voice command — for methods (a) and (b). The paper sheet instructed the points at which edit-menu and format-menu commands such as "new line" and "center justify" should be used while dictating. To evaluate whether the subjects preferred to use the speech-shift function, we then measured the usage frequency of method (d) under the condition that a subject could freely use any of the four methods according to personal preference. After the testing under both sets of conditions, the subject was asked to complete a subjective questionnaire.

We found that methods (a) and (d) took the shortest required time (almost the same) on average; method (c) took slightly (4%) longer, and method (b) took much (20%) longer. The relative usage frequency of method (d) among the four methods was 79.8%. These results showed that method (d) was efficient, compared to the other methods, and that the subjects preferred to use the speech-shift function even when they could choose otherwise. The questionnaire results indicated that method (d)

was the most preferred according to pairwise comparisons of the four methods, that the speech-shift function was easy to use and labor-saving, and that 85% of the subjects wanted to use the speech-shift function in the future.

7. Conclusion

We have described a new speech interface function *"speech shift,"* which judges whether the pitch of each utterance is normal or high in order to enable a user to specify the intended speech-input mode without using any other device — i.e., simply by intentionally changing voice pitch. To make this judgment robust for various users, we use a filled pause to estimate the pitch of each user's natural voice. The speech-shift function was implemented in a voice-enabled word processor system and proved to be an effective means of entering voice commands during dictation without explicitly switching between the dictation and voice-command modes.

The speech-shift function is a general idea that frees a user from having to take care of the current speech-input mode. We therefore plan to apply this idea to other voice-enabled applications. In addition, the idea of making full use of intentional nonverbal speech information in interface functions originated from research on "speech completion" [8, 9], which was followed by this research on "speech shift." Our future work will also aim at further developing this concept.

8. References

[1] Alex Waibel, "Prosodic knowledge sources for word hypothesization in a continuous speech recognition system," In *Proc. of ICASSP 87*, pp. 856–859, 1987.

[2] Andreas Stolcke, Elizabeth Shriberg, Dilek Hakkani-Tür, and Gökhan Tür, "Modeling the prosody of hidden events for improved word recognition," In *Proc. of Eurospeech '99*, pp. 311–314, 1999.

[3] Keikichi Hirose and Koji Iwano, "Detection of prosodic word boundaries by statistical modeling of mora transitions of fundamental frequency contours and its use for continuous speech recognition," In *Proc. of ICASSP 2000*, pp. 1763–1766, 2000.

[4] Elmar Nöth, Anton Batliner, Andreas Kießling, Ralf Kompe, and Heinrich Niemann, "VERBMOBIL: The use of prosody in the linguistic components of a speech understanding system," *IEEE Trans. on Speech and Audio Processing*, 8(5), September 2000.

[5] Masataka Goto, Katunobu Itou, and Satoru Hayamizu, "A real-time filled pause detection system for spontaneous speech recognition," In *Proc. of Eurospeech '99*, pp. 227–230, 1999.

[6] Masataka Goto, Ryo Neyama, and Yoichi Muraoka, "RMCP: Remote music control protocol — design and applications —," In *Proc. of Intl. Computer Music Conf.*, pp. 446–449, 1997.

[7] Tatsuya Kawahara, Akinobu Lee, Tetsunori Kobayashi, Kazuya Takeda, Nobuaki Minematsu, Katsunobu Itou, Akinori Ito, Mikio Yamamoto, Atsushi Yamada, Takehito Utsuro, and Kiyohiro Shikano, "Japanese dictation toolkit — 1997 version —," *J. Acoust. Soc. Jpn. (E)*, 20(3):233–239, 1999.

[8] Masataka Goto, Katunobu Itou, Tomoyosi Akiba, and Satoru Hayamizu, "Speech completion: New speech interface with on-demand completion assistance," In *Proc. of HCI International 2001*, volume 1, pp. 198–202, 2001.

[9] Masataka Goto, Katunobu Itou, and Satoru Hayamizu, "Speech completion: On-demand completion assistance using filled pauses for speech input interfaces," In *Proc. of ICSLP 2002*, pp. 1489–1492, 2002.