VOICE RECOGNITION FOCUSING ON VOWEL STRINGS ON A FIXED-POINT 20-MIPS DSP BOARD

Yukikuni NISHIDA, Yoshio NAKADAI, Yoshitake SUZUKI and Tetsuma SAKURAI

Speech and Acoustics Laboratory,
NTT Human Interface Laboratories
1-1 Hikarino-Oka, Yokosuka, Kanagawa,
239-0847 JAPAN

Toshihide KUROKAWA and Hirokazu SATO

Multimedia Business Development Headquarters
Speech & Audio Technology Center,
NTT Advanced Technology Corporation
Hakuei Bldg., 2-4-15, Nakacho, Musashino-shi,
Tokyo 180-0006 JAPAN

ABSTRACT

This paper describes a smart recognition system which performs character matching by replacing speech parameters with the five Japanese vowels and a few consonant categories. The proposed algorithm can make speaker-independent voice recognition. The algorithm has an advantage over the conventional speaker-independent word recognition system because it can reduce the required memory to about 0.5% of the conventional algorithm for storing the reference templates and for the instruction set, and can be performed even in a low-speed processor. We implemented this recognition algorithm in a fixed-point, 20-MIPS digital signal processor board with a 9-k x 16-bit on-chip RAM. Recognition experiments using 20 Japanese city names had a 90.3% accuracy. Such an accuracy is good enough for a voice control system.

1. INTRODUCTION

Voice recognition technology is now installed on some personal computers, telephones and other equipment. On a personal computer, the voice recognition is generally speaker-independent. However, in most other equipment, the voice recognition is speaker-dependent. Speaker-independent voice recognition requires a huge amount of memory. This means speaker-independent voice recognition on a processor chip with at most 30k-words of internal memory size is extremely difficult to achieve.

Taking the SPLIT (Strings of Phoneme-Like Templates) method [1] as an example of a speaker-independent isolated word recognition algorithm, memory is mainly consumed as a result of a distance matrix, which stores the spectral distance between each input word frame and each phoneme-like template. If only vowels are used as the phoneme-like template, the size of the phoneme-like template and the size of the distance matrix can be reduced. If only the character that contains the most similar phoneme to the input voice frame is stored instead of the spectral distance between each input voice frame and each phoneme-like template, the distance matrix can significantly be reduced in size. The amount of the memory can be expected to be reduced by using the above idea.

The authors wish to thank Y.Nishino for his guidance.

Vowel voice recognition was the first to be studied in voice recognition research [2],[3]. Vowels are steady timewise when compared with other phonemes. In addition, because vowel has a fundamental pitch frequency and its harmonics, the SNR of the vowels is better, and the shapes of an individual spectrum for vowels are more distinguishable than those for other phonemes. Therefore, vowels are suitable phonemes for voice recognition even in a noisy environment. Moreover, for a recognition task involving 94 underground station names in Tokyo, it has been shown that 92 of these station names can distinguished if the vowel part of each station name can be correctly recognized [4].

This paper proposes a voice recognition algorithm in which voice strings with distinguished vowels and non-distinguished (roughly categorized) consonants are matched with reference templates. The algorithm was implemented in only one-chip general-purpose fixed-point digital signal processor (DSP). The voice recognition performance was evaluated on the DSP board.

2. CHARACTER STRING GENERATION ALGORITHM FOCUSING ON VOWELS

2.1. Basic Algorithm

Figure 1 shows the voice recognition processing. The input voice is analyzed at a frame length of 32 msec and frame rate of 16 msec. The log power, the zero-cross counts and LPC cepstrum up to tenth order are calculated for the input voice. In the character string output part, the input voice is converted into a total of eight characters, that is: pause /*/, unvoiced fricative /S/, Japanese five vowels /a/,/i/,/u/,/e/,/o/, and nasals /N/, with each analysis frame. Table 1 shows samples of the character string produced from this string output algorithm. Next, in the character string compression part, the derived consecutive characters are rewritten in one character string. The compressed strings are also shown in Table 1. In the pattern matching part, the simplified character string obtained from character string compression part is collated with the character string of reference templates.

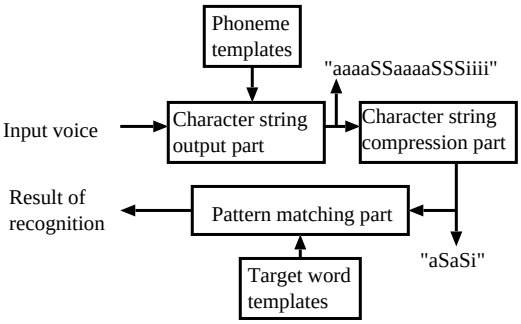

Figure 1: Block diagram of our speaker-independent word recognition algorithm.

Table 1: Example of character output

Pronunciation	output of character string output part	output of character string compression part
Hachinohe	aaaaa*SSiiiNNNooooSSSeeeeee	a*SiNoSe
Kesen'numa	SeeeSSSeeeNNNNNNNNuuuNNaa	eSeNNuNa
Yukuhashi	iiuuuu**uuuSSaaaaSSSSSiiii	iu*uSaSi

2.2. Initial Classification

The classification of eight phonemes is performed in two steps. The first classification step categorizes the input speech by using a two dimensional plane in which the log power is plotted along the x-axis and zero-cross counts are plotted along the y-axis. Figure 2 shows the result of plotting the log power and zero-cross counts of unvoiced fricative and vowels in the plane. Figure 2 shows that although the two categories overlap, they can still be distinguished most of the time.

The character /*/ corresponds to stops. If the log power in the analysis frame does not exceed the threshold calculated using the log power of the surrounding noise measured in non-voice period, character /*/ is output. The character /S/ corresponds to the voiceless fricative. When the input voice is judged to be a unvoiced fricative, the character /S/ is output.

A linear discriminant function, which has a low Bayes error, is used as a discriminant function to distinguish between the unvoiced fricative and the vowels. If the discriminant function for a rough classification of the two-dimensional plane is drawn, it becomes as shown in Figure 2. Thus, The input frame was classified into character /*/, character /S/, and other characters by this classification step.

2.3. Vowel Distinction

The second step is the classification of vowels. As non-categorized phonemes in the first step are a voiced consonant and vowels, they are classified into the characters of five typical Japanese vowels /a/,/i/,/u/,/e/,/o/ and the character /N/, which corresponds to the nasal.

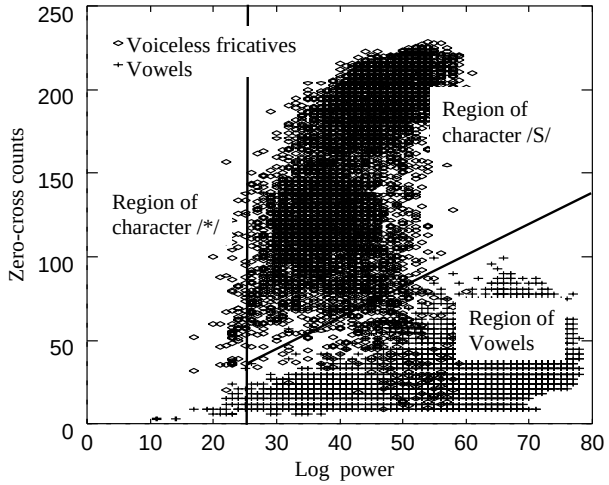

Figure 2: Distribution of voiceless fricatives and vowels in the (frame power, zero-cross counts)plane, and the discriminant diagram for classifying pause, voiceless fricatives and vowels

The spectral distance between the input frame and character reference template is then calculated. The character for which the distance value is the closest is output as a character of the input frame. The spectral distance measure is decided based on the distance measure evaluation experiment for each character. For the experiment, we used the cepstrums between first order and tenth order as the characteristic parameters. The test was done on three different distance measures: the Euclidean distance, the Mahalanobis distance, and the Bayes decision rule.

The reference templates of these tests are the mean cepstrums of the five vowels and the nasal /N/ for 20 male and 20 female voices a total of 12 mean cepstrums. The common full covariance, which is the averaged full covariance for each spoken character, was used as the probability distance measure calculation. This is because the on-chip memory of the DSP used as a test platform was not able to store the full covariance data for every character for male and female voices.

The Bayes decision rule gave a 78.8% phoneme recognition rate, which was the highest score of the three distance measures. Thus, we used the Bayes decision rule in our system.

Up to this point, the distinguished phonemes are transformed into a character in each frame, and output from the output part shown in Figure 1.

2.4. Character String Compression

To decrease the effect of transition from consonants to vowels, a discontinuous character is removed in the character string output by the previous output part. In addition, to correct the weight of each character and to reduce the amount of operations for the pattern matching, a consecutive character is rewritten as one character. The target

characters for the deletion are vowel characters and the unvoiced fricative character /S/. The stopped consonant character /*/ is not deleted because its duration time is shorter than vowels and unvoiced fricatives. After processing those two steps, if a number of consecutive characters is larger than a certain threshold number, one character is added. We introduced this measure to prevent character string /aa/ from being compressed into character /a/ through deletion, say for example when the character /r/ is deleted when /ara/ is input. The extension to two characters is used to express a long duration vowel.

3. CHARACTER STRING GENERATION OF REFERENCE TEMPLATES

The character string used for reference templates is generated by the following automatic rules. From the character string, which is written in Japanese syllabic characters, any unvoiced stops are changed into the character /*/, any unvoiced fricatives are changed into the character /S/, nasals are changed into the character /N/, voiced stops are changed into the character /u/, voiced fricatives are changed into the character /N/, the semi-vowel /y/ is changed into the character /i/, the semi-vowel /w/ is changed into the character /o/, and the vowels of long duration are changed into two characters.

However, because of coarticulation effects, certain classification errors appeared in character strings. For instance, the Japanese word "bibai" (city name) is almost always output as the character string "ui*uaei" by our character string output part. By using the transformation rule /ai/ $\rightarrow$ /aei/, we are able to make a character string template of reference templates.

4. CHARACTER STRING MATCHING METHOD

DP (Dynamic Programming) matching was used to evaluate the degree of correspondence between character strings of the input frame and reference templates. It is necessary to reduce the influence of the character omission or insertion. The distance between input character string and template characters is given by

$$D(t) = G(Imax, Jmax)/(Imax + Jmax), \qquad (1)$$

where $Imax$ and $Jmax$ are the lengths of the input character string and template character string.

The intermediate distance values are stored in a register $G(i,j)$, which is given by

$$G(i,j) = min \begin{bmatrix} G(i-1,j) \\ G(i-1,j-1) \\ G(i,j-1) \end{bmatrix} + g(c_i^R, c_j^T) \qquad (2)$$

$$1 \leq i \leq Imax, \quad 1 \leq j \leq Jmax. \qquad (3)$$

Here, $g(c_i^R, c_j^T)$ is obtained by referring to the distance value table between each character. The c_i^R and c_j^T are the i-th input character and the j-th template character, respectively.

The character distance values are obtained by

$$g(c_i^R, c_j^T) = -log(P(c_j^T|c_i^R)), \qquad (4)$$

where the a posteriori probability $P(c_j^T|c_i^R)$ is calculated by using the probability of the observed character when the phoneme of each character is input to the identifier. As there are eight characters for discriminating the input sound, i.e., five vowel characters and /N/,/*/ and /S/, the distance table contains 64 words at this time.

5. IMPLEMENTATION ON DSP BOARD

The system described above was implemented on a C5xDSK [5] that is available on the market. Figure 3 shows the construction of the whole recognition system.

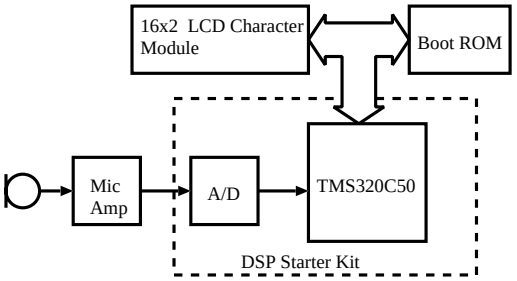

Figure 3: The structure of the processor implementation.

The TMS320C50 DSP and the 14-bit precision A/D converter including an anti-alias filter are on the C5xDSK. The DSP has 9k x 16-bit on-chip RAM for programs or data, a 16 x 16-bit parallel multiplier with a 32-bit product capability, and a 32-bit arithmetic logic unit with performance of 50 nsec. The recognition result is displayed on an LCD character module. The program to evaluate the proposed algorithm is stored in the boot ROM. Resetting the DSP copies the executable code from the boot ROM to the on-chip RAM. After that, it does not have to access the boot ROM again until it is reset.

The program, which included the phoneme table and LCD device driver of the algorithm, was 8k x 16-bit in size. Here, the memory space that we were able to use for character strings of reference templates was 1k x 16-bits. The amount of memory used for one reference template was 21 x 16-bits, so, the maximum number of reference templates is about 50.

Table 2 shows the comparison of memory size of proposed recognition system and a recognition system based on a Hidden Markov Model (HMM) [6], which is capable of speaker-independent voice recognition. The proposed voice recognition system was able to perform speaker-independent voice recognition in a memory size 0.5% that of the conventional recognition system based on HMM.

6. EXPERIMENT

To evaluate the effectiveness of the proposed recognition algorithm in speaker-independent conditions, we selected the first 10, 20 and 30 Japanese city names from among the 100 Japanese city names proposed by the Japan Electronic Industry Development Association as reference templates

Table 2: Comparison of memory size of the proposed recognition system and the large vocabulary recognition system based on the HMM when the number of the recognition words is 20; PROP - the proposed recognition system; HMM - recognition system based on HMM.

Processing part		PROP (KByte)	HMM (KByte)	Reduced size (%)
Analysis part	Instruction	12	200	6 %
	Work area	2	700	0.3 %
Matching part	Instruction	2	1,000	0.2 %
	Work area	0.1	1,200	0.008 %
The size of dictionary		0.9	540	0.2 %
Total		17	3,640	0.5 %

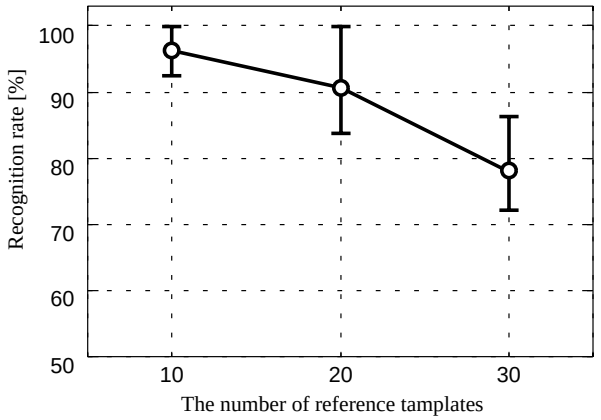

Figure 4: The range of the recognition rate relative to the number of reference templates, with the average indicated by a circle.

[7], we used speech data uttered four times each by 10 male and 10 female speakers. The phoneme string of the reference templates was made from important words in the ATR Japanese speech database [8].

Figure 4 gives the range of the recognition rate, which is the averaged recognition rate of a word uttered 4 times by each speaker for the number of the reference templates. The circles indicate the average recognition rates for all speakers. The number of reference templates for which practical recognition rates can be obtained was less than 20 words. However, for voice controlling telephones, toys, etc., we think the ability to recognize 10 words simultaneously is sufficient because the number of commands in a particular category can be limited to less than 10 words.

Due to the memory size, the proposed recognition system can recognize up to 50 words with one-chip DSP. The recognition time was about 0.3 sec. for 20 word recognition, from end of an utterance to output of the recognition result.

7. CONCLUSION

We developed a smart voice recognition algorithm, which can be installed on a DSP board with a fixed-point 20-MIPS performance and 9k x 16-bit RAM. It distinguishes mainly vowels and unvoiced consonants through voice signal power and the zero-cross counts. The algorithm can reduce the size of memory and processor performance requirements for such an application by compressing the character string and using a posterior probability, which is stored in a table as the distance score between each character when the input character string and reference template character string is compared. A maximum of 50 words can be stored in the memory of the one-chip DSP.

We conducted speaker-independent voice recognition experiments using a vocabulary of 10, 20 and 30 Japanese city names to evaluate the proposed algorithm. When utterances by another 10 males and 10 females of 20 Japanese city names were used as voice input, the recognition rates were 90.3 % on average.

When we change recognition words, the reference template need not relearn: it is generated by putting recognition words into Japanese syllabic characters.

8. REFERENCES

[1] N. Sugamura, K. Shikano and S. Furui, "Isolated word recognition using phoneme-like templates," in Proc. Int. Conf. Acoust. Speech Sign. Process., pp. 723-726, Apr. 1983.

[2] James W. Forgi and Carma D. Forgie, "Result Obtained from a Vowel Recognition Computer Program," J. Acoust. Soc. Amer., vol. 31, pp. 1480-1489, Nov. 1959

[3] K.H. Davis, R. Biddulph and S. Balashek, "Automatic Recognition of Spoken Digits," J. Acoust. Soc. Amer., vol. 24, pp. 637-642, Nov. 1952

[4] T. Takara and S. Imai, "Vowel Recognition Based on Mel-Sone Spectrum," Trans. the Institute of Electronics and Communication Engineers, vol. J65-A, pp. 818-825, Aug. 1982.

[5] User's Guide: TMS320C5x DSP Starter Kit, TEXAS INSTRUMENTS 1996

[6] Y. Nakadai, Y. Suzuki and T. Sakurai, "A Speaker-independent Japanese Thousand Word Recognition CTI Board," Proc. AVIOS '98 (in publication)

[7] S. Itahashi, "A Japanese Language Speech Database," Proc. ICASSP 86, vol. 1, pp. 321-324, 1986

[8] Y. Sagisaka, K. Takeda, M. Abe, S. Katagiri, T. Umeda and H. Kuwabara, "A Large-Scale Japanese Speech Database," Proc. of the International Conference on Spoken Language Processing, vol. 2, pp. 1089-1092, 1990