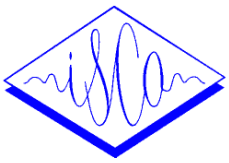


2nd International Conference on
Spoken Language Processing (ICSLP 92)
Banff, Alberta, Canada
October 12-16, 1992

HARDWARE IMPLEMENTATION OF REALTIME 1000-WORD HMM-LR CONTINUOUS SPEECH RECOGNITION

Akito NAGAI[1], *Kenji KITA*[1], *Toshiyuki HANAZAWA*[2], *Tadashi SUZUKI*[2], *Tomohiro IWASAKI*[2], *Tsuyoshi KAWABATA*[3], *Kunio NAKAJIMA*[2], *Kiyohiro SHIKANO*[4], *Tsuyoshi MORIMOTO*[1], *Shigeki SAGAYAMA*[1] and *Akira KUREMATSU*[1]

[1]ATR Interpreting Telephony Research Laboratories (2-2, Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-02 JAPAN)
[2]Mitsubishi Electric Corporation [3]NTT Basic Research Laboratories [4]NTT Human Interface Laboratories

ABSTRACT

This paper describes the design and development of the architecture of dedicated hardware which recognizes continuous speech sentences using a 1000-word vocabulary. It is based on the HMM-LR mechanism, which is an integration system of speech recognition and language analysis. Many real-time techniques using 33 DSPs for multi-processing were developed for the heavy computation of the trellis algorithms and state duration control. The paper gives performance data with results within a pipeline delay of approximately 2 or 3 seconds, independently of the length of the sentence.

1 INTRODUCTION

Accurate and efficient speech parsing is achieved through the integrated processes of speech recognition and language analysis. The HMM-LR [1] is a high-performance scheme which uses an efficient parsing mechanism, a generalized LR parser, driving an HMM-based speech recognizer directly without any intervening structures such as a phoneme lattice.

In order to accelerate speech recognition by the HMM-LR with a grammar of perplexity greater than 100, the authors have collaborated to design the many real-time techniques required to efficiently execute the calculation for the trellis algorithms and duration control. We also hope to evaluate the feasibility of a one-chip speech recognizer through the development of this hardware.

The hardware recognizes human speech at a rate approximately 100 times faster than equivalent software running on a 24 MIPS workstation.

On input of a sentence, phrase by phrase, it takes only 2 or 3 seconds to recognize the input speech, independently of the length of the sentence.

The major points for high-speed speech recognition are ; (1) *The pipeline process in the Front End Processor*, (2) *The parallel process in the HMM verifiers*, (3) *The distributive process in the phrase recognizers.*

This paper first outlines the HMM-LR system, second, describes real-time techniques for the efficient calculation for HMM-LR speech recognition with the trellis algorithms and state duration control, third, describes the architecture, fourth, evaluates the performance, and finally describes the incorporation to Japanese-English speech translation system.

2 HMM-LR ALGORITHM

We have been working on large-vocabulary continuous speech recognition that takes full advantage of grammatical constraints supplied by a syntactic language model. Our approach uses phone-based hidden Markov models (HMMs) for acoustic modeling, and generalized LR parsing [2] for dealing with grammatical constraints based on a context-free grammar. This integration, called HMM-LR, was successfully applied to Japanese phrase recognition including about 1,000 words [1][3].

In the HMM-LR, the generalized LR parser is used as a language source model for word/phone prediction/generation. First, the parser picks up all phones predicted by the initial state of the LR parsing table and invokes the HMM models to verify the existence of these predicted phones. The parser then proceeds to the next state in the LR parsing table. During this proceses, all possible partial parsers are constructed in parallel.

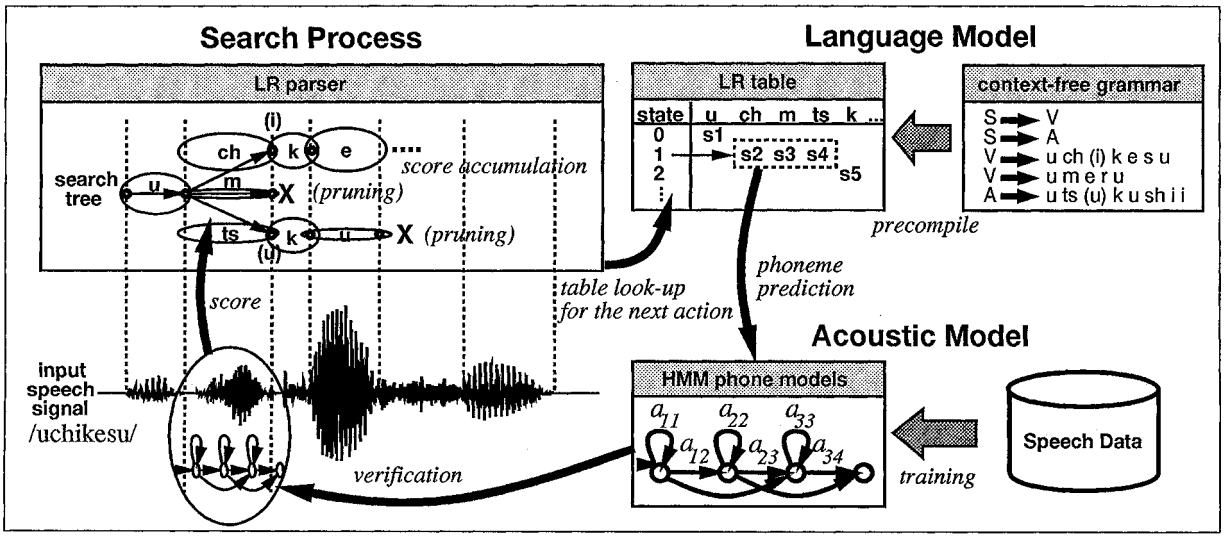


Figure 1. The HMM-LR continuous speech recognizer

The HMM phone verifier receives a probability array which includes end point candidates and their probabilities, and updates it using an HMM probability calculation. This probability array is attached to each partial parse. The parsing process proceeds in this way, and stops if the parser detects an accept action in the LR parsing table. In this case, if the best probability point reaches the end of the speech, parsing ends successfully.

Through the integrated process of speech recognition and language analysis, very accurate, efficient speech parsing is achieved avoiding the information loss due to signal-symbol conversion.

For more accurate speech recognition, HMM state duration control and fuzzy vector quantization are used. Table 1 shows the specification of the HMM-LR system. The speech is sampled at 12 kHz, pre-emphasized with a transfer function of $(1-0.97z^{-1})$, and windowed using a 256-point Hamming window every 9 msec. Then, 14-order LPC analysis is carried out.

Table 1. the specification of the HMM-LR system

A/D	16 bit, 12 kHz sampling
frame length (period)	21.3ms (9.0ms)
LPC analysis	14-order (Hamming window)
analysis parameter	16-order auto-correlation
	16-order LPC cepstrum
	power
	16-order Δ cepstrum
	Δ power
codebook (size)	WLR (256)
	Δ cepstrum (256)
	Δ power (64)
fuzzy VQ	fuzziness 1.5, 6-nearest neighbor
# HMM	3-loop 38, 1-loop 56
algorithms	trellis, state duration control
# rules	1500
# LR states	4500
max. of phrase length	256 frame
max. of beam width	256

Fuzzy vector quantization (Fuzzy VQ) is done based on k-nearest neighbors (k=6) and fuzziness 1.5. HMM phone models are based on discrete HMM; output probability is based on 3 VQ codebooks (WLR [4], difference cepstrum coefficients and difference power). A three-loop model is for consonants and a one-loop model for vowels. To reduce search space, a beam-search technique is used.

3 TECHNIQUES FOR THE REAL-TIME EXECUTION

In order to realize realtime processing speed for the HMM-LR continuous speech recognition without reducing the recogniton accuracy, the HMM-LR recognition process is mainly divided into the front end process as a common unit and the LR parsing process with HMM trellis calculation as a parser unit as shown in Figure 2. A frame-synchronous pipeline processing is carried out on input speech in the common unit. Two kinds of parallel processing and a distributive processing are carried out on each phrase utterance as a unit. The following techniques were used;

3.1 The Pipeline Process in the Front End Processor

The front end function of the HMM-LR is divided into three modules, namely, the speech analysis module for speech sampling and 14-order LPC analysis, a fuzzy vector quantization module, and a calculation module for the HMM output probability. These processes are pipelined frame-synchronously so as to finish the execution in each module within one frame period, and simultaneous analysis with speech input is realized.

3.2 The Parallel Process in the Phrase Recognizer

Two kinds of parallel processing about a phrase utterance as a recognition unit are done in a parser unit; one is for a global mechanism of HMM-LR between the LR parser and the HMM trellis verifier, and another for the HMM trellis calculation.

In the first strategy, two parallel operations for dividing the LR parsing processes are possible by modifying the control of parsing hypotheses.

In the original HMM-LR algorithm, a data structure called a *cell* is used for a parsing hypothesis. A cell is a structure with information about one recognition candidate. The following are kept in the cell: 1. *LR parsing stack*, with information for parsing control, 2. *Probability array*, which includes end point candidates and their probabilities.

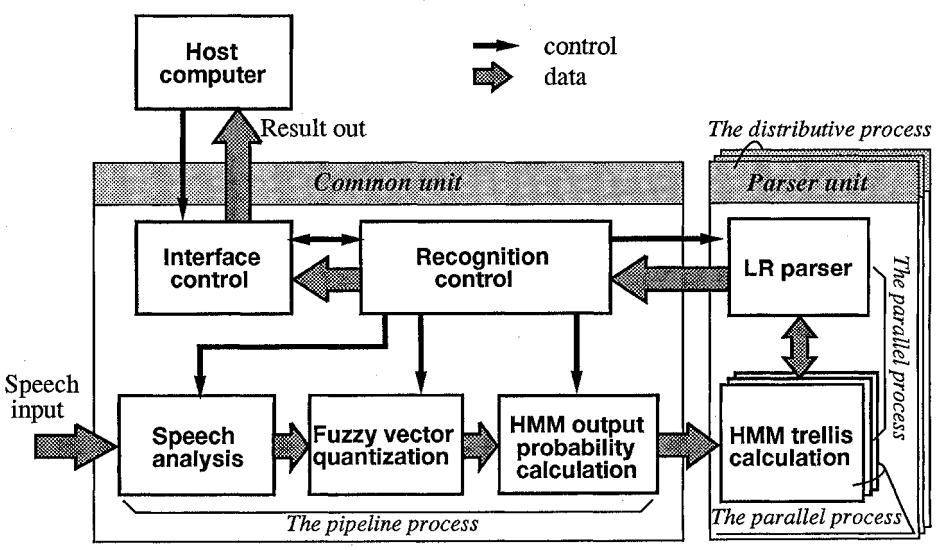


Figure 2. The functional structure

The LR parser first predicts phones in the LR parsing table, and grows parsing trees with (1) *shift* action or (2) *reduce* action by the push/pop operation on the LR parsing stack of each cell. The *shift* action leads to the HMM verification, and the *reduce* action is related only to LR parsing operation. Second, the predicted phones are verified by (3) HMM trellis calculation and the probability array is updated. Finally, (4) the beam search is done for pruning.

For the purpose of independently controlling the LR parsing stack and the probability array, they are separated in the functional modules. Consequently, (2) and (3) can be operated simultaneously. Moreover, by operating *shift* action during the transfer time of cell's data to the beam search function, (1) and (4) also can be executed in parallel.

The second strategy for realtime execution is multi processing of the HMM trellis computation.

Table 2 shows the computational ratio of each principal execution time to all the HMM-LR processing time. The input data to HMM-LR is the result of the fuzzy vector quantization. This ratios are the averages for 10 phrases. According to the results, the trellis calculation with a probability addition using logarithmic representation amounts to about 65% of the total HMM-LR computation. This calculation can be executed in parallel by sharing the information of both the output probability in each state transition of all HMMs and the table for logarithmic addition. 10 parallel processing units enable reduction of the calculation rate to 6.5%; 20 units reduce the rate to 3.3%.

Table 2. HMM-LR processing ratios (%)

trellis calculation	51.0
probability addition	14.2
cell copying	3.4
output probability calculation	2.8
beam search	2.1
LR parsing	0.5

3.3 The Distributive Process in the Phrase Recognizers

The input speech to the HMM-LR hardware is a sentence consisting of several phrases. If the processing time of a phrase takes more than the phrase duration in the case of continuous utterance, the accumulation of recognition delay occurs. Therefore, by controlling more than one phrase recognizer i.e. parser unit in parallel and simultaneously recognizing phrases in each parser unit, the reduction of this accumulative delay is realized and the total response time of a sentence is accelerated independently of the length of the sentence.

3.4 The real time operation utilizing the constraints of phonemic durations

This operation is for the purpose of real time execution with no wait for finishing input utterance. By exploiting the information about each phonemic duration model, the possible verifications are first done on the phonemic hypotheses for speech units already completely input.

4 HARDWARE IMPLEMENTATION

In order to realize the real time operations mentioned above and to recognize speech with sufficient precision for maintaining the accuracy, the architecture was designed as follows:

4.1 The Architecture

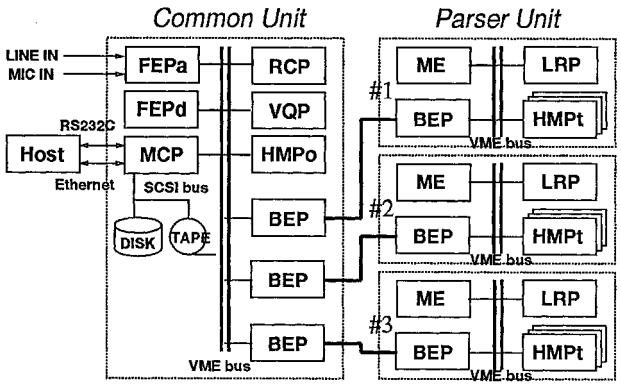
The hardware mainly consists of a common unit and a maximum of three parser units as shown in Figure 3. Each functional module is constructed on the substrate for VME bus. To improve data communication in the VME bus, two techniques are adopted; simultaneous conveyance of identical data to multiple DSP boards, and continuous group conveyance of data to reduce bus access overheads.

In the common unit, MCP is for an interface control. UNIX runs on it and communicates with host computer by ethernet or RS232C. RCP is for the HMM-LR recognition control on other modules. FEPa is an analog-to-digital converter, and FEPd is a speech analyzer. In the parser unit, ME is a memory for the communication between LRP and HMPt. Each parser unit has fifteen slots for HMPt substrates.

33 digital signal processors (DSPs) and 2 micro processing units (MPUs) are used as follows: 1 MPU (MC68030, clock frequency 25 MHz, Motorola) for MCP and RCP, 1 DSP (mSP2, floating-point/integer multiplier, machine cycle 75ns, Mitsubishi Electric Corp.) for FEPd, 32 DSPs (TMS320C30, floating-point/integer multiplier, machine cycle 60ns, Texas Instruments) for VQP(1), HMPo(1) and HMPt(30), 1 MPU (MC68030, clock frequency 32 MHz, Motorola) for LRP.

4.2 The process flow

The main process flow is outlined in Figure 2. Abbreviations like *RCP* are explained in Figure 3. After booting the system, first, RCP begins to control FEPa, FEPd and VQP in pipeline, and starts the speech detecting process analyzing input signal. MCP sends the starting instruction of HMM-LR recognition to RCP at the moment the host computer requires. When a speech signal is detected in RCP, it transfers the vector quantization parameters to HMPo and orders LRPs to start the HMM-LR parsing in the parser units. The recognition results from LRPs are returned to the host computer through RCP and MCP.



MCP : Master Control Processor
RCP : Recognition Control Processor
FEPa : Front End Processor for Analog signal
FEPd : Front End Processor for Digital signal
VQP : Vector Quantization Processor
HMPo : HMM Processor for Output probability
HMPt : HMM Processor for trellis algorithm
LRP : LR Parser
ME : Message Exchanger
BEP : Bus Expander/repeater

Figure 3. Hardware architecture

5 PERFORMANCE

The 1000-word *International Conference Registration* task is described by context free grammars including 1407 rules with 5.9 phonetic perplexity and more than 100-word perplexity (Table 3). The experiment was carried out under the conditions in Table 4. The length of a test sentence uttered phrase by phrase was about 30 seconds including silences between the phrases. The real speech length was 16.6 seconds consisting of 23 phrases. Two parser units with search beam width 100 were used.

Table 5 shows the response times from the end of the input speech to the finished recognition. The hardware finished whole sentence recognition in about 2 seconds, and it was approximately 100 times faster than equivalent software running on a 24 MIPS workstation. This processing time of the hardware included the waiting time for inputting the next phrase in the silence duration.

The results of demonstrations for visitors proved that this hardware recognized test sentences of variable length within 2 or 3 seconds independently of the length.

Table 3. Grammar

vocabulary	1035
#rules	1407
phonetic perplexity	5.9
word perplexity	> 100

Table 4. Conditions

speaker	a male announcer
sentence length	30 sec.
total length of phrases	16.6 sec.
#phrases	23
input source	line input from a DAT
beam width	100
#parser unit	2

Table 5. Response times (sec)

the hardware	DECstation 5000 (24 MIPS)
2	209 *

* : 29.5 for VQ, 179.8 for HMM-LR

6 INCORPORATION TO JAPANESE-ENGLISH SPEECH TRANSLATION SYSTEM

The hardware has been already incorporated into SL-TRANS2 [5][6], which is an experimental spoken language translation system to demonstrate the feasibility of an automatic interpreting telephony (Figure 4).

This system recognizes spoken Japanese sentences, translates them into English and outputs synthesized English speech. It is composed of four subsystems; an HMM-LR continuous speech recognizer, a sentence candidate generator using an inter-phrase grammar, a dialogue translator and an English speech synthesizer using DECtalk. These subsystems are integrated by a system controller.

In addition to HMM-LR, a hardware unit for speaker adaptation was also implemented in SL-TRANS2 based on a fuzzy codebook mapping algorithm [7], not treated here. The algorithm of this speaker adaptation unit required 25 words utterances for training.

Recently, we have proposed a new adaptation algorithm based on *Vector Field Smoothing (VFS)* [8][9], which realizes the adaptive training using only 10 words for equivalent performance to the adaptation based on fuzzy codebook mapping. This algorithm enables the training to finish in about one minute on a 76-MIPS workstation.

We are planning to implement the VFS algorithm into the speaker adaptation unit, which will require fewer words of a new speaker and less time for training than a conventional HMM-LR system.

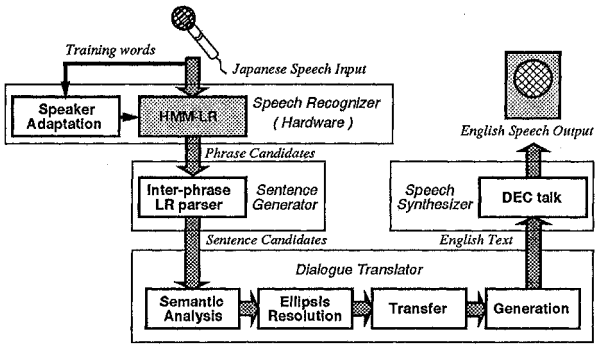


Figure 4. SL-TRANS2 speech translation system

7 CONCLUSION

The authors designed the architecture of dedicated hardware for HMM-LR; integrating processes of speech recognition and language analysis. Using 33 DSPs, the pipeline process in the Front End Processor, The parallel process in the HMM verifiers and the distributive process in the phrase recognizers were developed. As a result, the HMM-LR hardware has realized realtime 1000-word continuous speech recognition with word perplexity greater than 100. By incorporating it into SL-TRANS2, the processing time of Japanese-English speech translation has decreased successfully.

REFERENCES

[1] K. Kita, T. Kawabata and H. Saito, "HMM Continuous Speech Recognition Using Predictive LR Parsing," Proc. ICASSP89, pp. 703-706 (May 1989).

[2] M. Tomita, "Efficient Parsing for Natural Language: A Fast Algorithm for Practical Systems," Kluwer Academic Publishers (1986).

[3] T. Hanazawa, K. Kita, S. Nakamura, T. Kawabata and K. Shikano, "ATR HMM-LR Continuous Speech Recognition System," Proc. ICASSP90, pp. 53-56 (April 1990).

[4] M. Sugiyama and K. Shikano, "LPC Peak Weighted Spectral Matching Measures," Trans. of IECE, Vol.J64-A, No.5, pp.409-416 (1981.5) (In Japanese).

[5] T. Takezawa, K. Ohkura, T. Morimoto, S. Sagayama and A. Kurematsu, "SL-TRANS 2: an Experimental System for Translating Japanese Speech to English," The Acoustic Society of Japan Fall Meeting Proc., 1-5-24, pp.47-48 (1991.10) (In Japanese).

[6] T. Morimoto, M. Suzuki, T. Takezawa, G. Kikui, M. Nagata and M. Tomokiyo, "A Spoken Language translation system : SL-TRANS2," Proc. of COLING'92.

[7] S. Nakamura, S. Hamaguchi, A. Nagai, K. Shikano, A. Tanaka, S. Sagayama and A. Kurematsu, " Development of VQ-based Speaker Adaptation System," The Acoustic Society of Japan Fall Meeting Proc., 3-5-6, pp.101-102 (1991.10) (In Japanese).

[8] H. Hattori and S. Sagayama, " Evaluation of Transfer Vector Field Smoothing Speaker Adaptation Method on Japanese Phrase Recognition," The Acoustic Society of Japan Spring Meeting Proc., 2-Q-15, pp.187-188 (1992.3) (In Japanese).

[9] H. Hattori and S. Sagayama, "Vector Field Smoothing Principle for Speaker Adaptation," ICSLP'92 (1992.10).