# PARALLEL LVCSR ALGORITHM
# FOR CELLPHONE-ORIENTED MULTICORE PROCESSORS

*Shin-ya ISHIKAWA     Kiyoshi YAMABANA     Ryosuke ISOTANI     Akitoshi OKUMURA*

Media and Information Research Laboratories, NEC
s-ishikawa@dg.jp.nec.com

## ABSTRACT

A parallel Large Vocabulary Continuous Speech Recognition (LVCSR) algorithm for cellphone-oriented multicore processors is proposed. We introduce an acoustic look-ahead and blockwise computation to our compact LVCSR algorithm, in order to distribute its computational load to multiple CPU cores. We implement the proposed LVCSR algorithm on an evaluation board of a cellphone-oriented three CPU core chip, and show real-time proccessing of stand-alone LCVSR on cellphones can be achived with recognition vocabulary of about 50,000 words. We also implement a speech-input text retrieval system using the proposed LVCSR on the same evaluation board, and confirm the ability of the proposed LVCSR algorithm to provide comfortable responses to query sentences spoken to a cellphone, without requiring any outside resources.

## 1. INTRODUCTION

Large Vocabulary Continuous Speech Recognition (LVCSR) is an expected technology for portable devices with limited number of keys such as cellphones. It provides an easier, simpler and faster text input method via speech compared to the conventional method via the limited number of keys which requires complex key strokes. For example, in writing e-mail or short messages on cellphones, the users have only to speak the sentences which they want to write, while in the conventional method they have to manipulate the numeric keypad on cellphones character by character. For another example, recent cellphones are getting so complicated day by day that it is not easy for users to find the function that they need. LVCSR enables the users to input multiple key words to search for the function in one breath, by speaking a query sentence to cellphones. However, LVCSR on cellphones has been impossible because the computational performance of cellphone processors is limited due to the limited capacity of cellphone batteries.

Server-client architecture is one of the solutions to the problem of the processor performance and battery capacity. In our previous work [1], we implemented and evaluated a speech-input and text-output retrieval system that can be used on cellphones by using LVCSR, WEB, and text retrieval servers. However, a network connection must be available to use this system and even when it is available users have to pay for it. In this sense, a stand-alone LVCSR on a cellphone has yet to be achieved.

We have also developed a compact LVCSR system, and have applied it to a Japanese/English bi-directional speech-to-speech translator on a PDA [2]. This translator performs LVCSR on StrongARM 206MHz, but still a few seconds are needed until users see the recognition result after they finished speaking to the translator. Better computational performance and lower power consumption are needed to achieve real-time recognition on cellphones.

Recently a cellphone-oriented processor chip with both high performance and low power consumption was announced [3]. The chip has three ARM9 CPU cores with relatively low operation frequency of 200MHz, which is advantageous in achieving lower power consumption than one CPU chip that has three times as high frequency, i.e. 600MHz. Because this is powerful enough for the compact LVCSR algorithm, and the power consumption is low enough for cellphones, the problem is whether we can divide the LVCSR algorithm into three parts and execute them by three CPU cores in achieving stand-alone LVCSR on cellphones.

In this paper, we propose a parallel LVCSR algorithm for multi-processor systems. We implement the proposed algorithm on a cellphone-oriented multicore chip, and evaluate whether real-time operation of LVCSR has been achieved. The recognition rate is evaluated in a travel conversational task. We also implement a microphone-input application system of the proposed LVCSR, which retrieves relevant passages in the users' manual of the cellphone using the recognition result as the query, and shows the 10 best retrieval candidates on the screen.

In the following sections we describe the baseline LVCSR system, the proposed parallel algorithm, the evaluation, and the speech-input text retrieval system.

## 2. BASELINE LVCSR SYSTEM

Here we describe a baseline compact LVCSR system applied to the PDA translator [2]. This system is oriented to

1-4244-0469-X/06/$20.00 ©2006 IEEE

portable devices such as PDAs, and thus works with limited computational resources.

### 2.1. Decoder

The decoder performs frame synchronous Viterbi beam search on a lexical prefix tree [4]. The lexical tree consists of all words that can be recognized by this system, and each word is compactly stored by expressing them in context-independent phonemes. Each of the phonemes needs only one byte in many cases because the number of different phonemes is less than 256. The decoder looks up triphone HMMs by a triphone index which is dynamically calculated from three successive phonemes in the decoding process. In each frame the hypotheses that reach the end of the words in the lexical tree are stored as “word ends” in the word end table, and finally all the word ends are output as a word graph at the end of the utterance. Word unigram probability is given as Language Model (LM) look-ahead values to the word hypothesis on the lexical tree, and at the end of the word that is replaced by bigram probability. The word graph is tracked back when the search reaches the end of each utterance, and the best recognition result candidate is obtained. It is also possible to add a second pass that processes the word graph and produces better recognition results. The memory consumption that depends on utterance length is suppressed by the garbage collection of the word ends.

The baseline LCVSR works on StrongARM 206MHz, but still a few seconds are needed until users see the recognition result after they finished speaking to the system.

### 2.2. Acoustic model and distance calculation

Triphone HMMs with tree based state clustering on phonetic context are used as the acoustic model. The state emission probability is represented by Gaussian mixtures with diagonal covariance matrices. We use three techniques to reduce the amount of memory and computation required for the acoustic model. The first is Gaussian reduction based on the MDL criterion [5], which efficiently removes redundant Gaussians in the states. The second is the global tying of the diagonal covariance matrices of Gaussian mixtures. The third, high-speed calculation of emission probabilities [6], achieves more than ten times faster computation with the least accuracy loss.

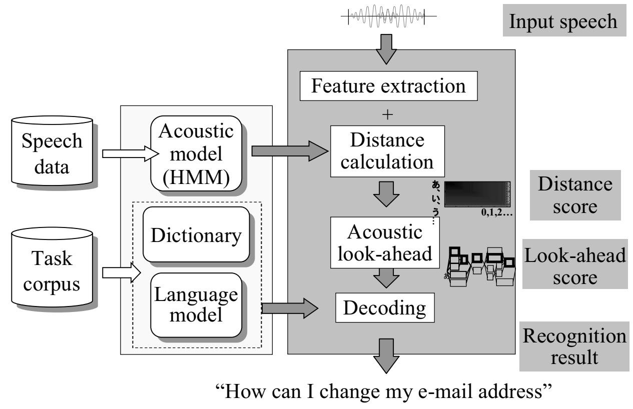


Figure 1: LVCSR with acoustic look-ahead

### 2.3. Language model

The language model is composed of bigram and class bigram probability entries estimated from a corpus of thousands of sentences. The class is based on the parts of speech of the Japanese language. The probability value is quantized to minimize memory consumption.

## 3. PROPOSED PARALLEL LVCSR ALGORITHM

### 3.1. Introduction of acoustic look-ahead

The baseline LVCSR algorithm consists of three steps: 1) feature extraction, 2) distance calculation between HMM and extracted feature vector, and 3) decoding using the distance scores, language model, and word dictionary stored in the lexical prefix tree.

One may think the system can be easily parallelized by executing the three steps with three CPU cores respectively, but this doesn’t make full use of the computational resources because the amount of computation for the decoding accounts for more than 60% of all the LVCSR computation, which is much greater than that for the other two steps. For better parallel processing, better balance among the three steps is necessary. In this sense the computational load for the decoding should be distributed to multiple CPU cores, though this is not easy because the word graph composed of preceding word ends is shared and constantly accessed by most of hypotheses in the decoding process.

Our first proposal is to introduce an acoustic look-ahead to distribute the computational load of the decoding to two new successive steps.

Acoustic look-ahead (first step) performs phoneme-level backward matching of the input utterance, from the end of the utterance back toward the beginning of the utterance. The backward matching suitably calculates each phoneme score in each frame, which is the score of the most likely phoneme sequence that starts with the phoneme in the frame and ends in the end frame of the input utterance. The acoustic look-ahead uses neither a word dictionary nor a word n-gram language model.

In the following decoding computation (second step), the sentence hyopotheses are pruned efficently and drastically by using the result of the preceding acoustic look-ahead as the speculation score of each phoneme in each frame [7]. Fortunately, the amount of computation of the two is comparable not only to that of each of them but also to the sum of feature extraction and distance calculation. Thus the LVCSR algorithm can be seen to comprise three new successive procedures with comparable calculation amount: 1) the feature extraction + distance calculation, 2) the acoustic look-ahead, and 3) the decoding (see Fig. 1).

### 3.2. Introduction of blockwise successive computation

If these three successive procedures were all to forward the input utterance, the LVCSR could be parallelly executed with ease with the latency of one frame between each successive procedure. However, this is not possible because the acoustic look-ahead processes the utterances backward.

Therefore we introduce as the second proposal a blockwise successive processing to achieve parallel processing of the three procedures with the latency of one block length between each successive procedure. Figure 2 shows the whole idea of this proposal. An utterance is divided into frame blocks of a constant length. Three procedures are independently executed by three CPU cores respectively within each frame block, and on the boundary of the blocks each result is passed to the next CPU core that executes the next procedure. For example, first, CPU1 executes feature extraction + distance calculation for block1, and at the end of the block CPU1 passes to CPU2 the distance scores of block1. Then CPU1 keeps executing feature extraction + distance calculation for block2 while CPU2 executes the acoustic look-ahead for block1. CPU3 starts the decoding for block1 after CPU2 finishes the look-ahead for block1 and passes both the distance and look-ahead scores of block1 to CPU3. While CPU3 executes the decoding for block1, CPU1 works for block3, and CPU2 works for block2. The parallel processing continues in this way until the end of the utterance.

### 3.3. Memory consumption

The above-mentioned block data passed among the three CPU cores are stored in the shared memory of the CPU cores. Note that the memory consumption is greater than that of the baseline system due to the nature of this blockwise parallel algorithm through which CPU cores execute their jobs independently, and thus need their own block data in shared memory at the same time.

Local memory is faster than shared memory because the former is accessed with a CPU cache while this cannot be done for the latter on this multicore processor. Therefore, the acoustic model and the language model + word dictionary are stored in local memory because they are accessed, respectively, only by CPU1 and CPU3.

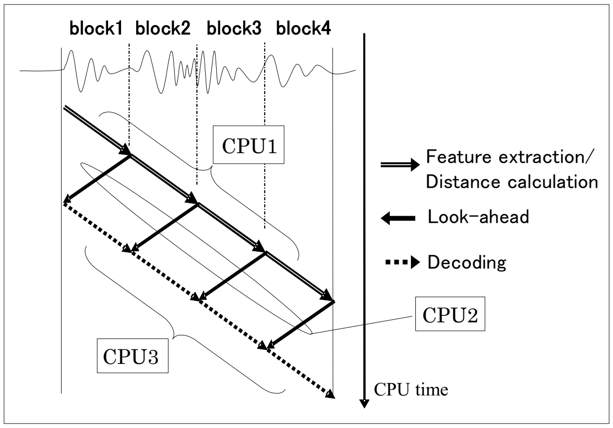


Figure 2

## 4. EVALUATION OF RECOGNITION SPEED AND ACCURACY

In this section we describe about the evaluation of the proposed LVCSR algorithm. In the following descriptions, “our proposed system” means the system which has the same configuration as the baseline system except for an introduction of our proposed parallel LVCSR algorithm.

### 4.1. Evaluation conditions

Both the baseline system and our proposed system were constructed and evaluated on an evaluation board of the cell-phone-oriented multicore processor. The processor has three ARM9 CPU cores, and the operation frequency is set to 150MHz. The baseline system uses only one CPU core, while the proposed system uses all the three cores. A speech signal is sampled at 11 kHz, with an MFCC analysis frame rate of 11ms. The acoustic model is speaker-independent but gender-dependent. The dictionary and the language model are those of the PDA speech-to-speech translator mentioned previously, and the number of recognition words is about 50,000.

Evaluation data for recognition speed and accuracy is 50 utterances and 1000 utterances respectively. The utterances are Japanese read-out sentences of the travel conversational task by five male speakers.

The evaluation utterances are stored in data files of RIFF format, and input to the LVCSR by an interval timer in order to simulate real-time input of speech from a microphone. This guarantees the system to take speech data from the RIFF files just as fast as it was uttered.

The block length in the proposed blockwise processing is set to 40 frames, which has been shown in a preliminary experiment to cause only a slight degradation in the recognition accuracy compared to the block of infinite length, i.e. the algorithm without the proposed blockwise proccessing.

### 4.2. Result

Table 1 shows the speed, the Percent Correct (PC), the Word Accuracy (WA), and the memory consumption of the baseline system and our proposed system. As for PC and WA, the difference between the proposed system and the baseline system is the existence of the acoustic look-ahead and the blockwise processing. These slightly degrade PC and WA, but they are still high enough for an LVCSR system. Speed (inclination) shows the ratio of the increase in processing time to the increase in the length of evaluation utterance. It is shown that the proposed system performs LVCSR within one Real Time (RT) except for the two parts of block latency shown in Fig. 2. One RT is the fastest in this system because of the interval timer mentioned above, and thus the proposed system may still have room for more calculation. As for memory consumption, the proposed system needs about 2M bytes more than the baseline system. This corresponds to the amount of memory for the acoustic look-ahead scores and the distance scores which are redundantly allocated in shared memory for independent processing by the three CPU cores.

Table 1

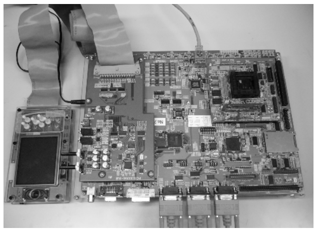
| | Baseline (1CPU) | Proposed (3CPU) |
|---|---|---|
| Speed(inclination) | 2.6RT | 1.0RT |
| Memory | 3.5Mbyte | 5.5Mbyte |
| PC, WA | 96.0%, 95.8% | 95.6%, 95.4% |

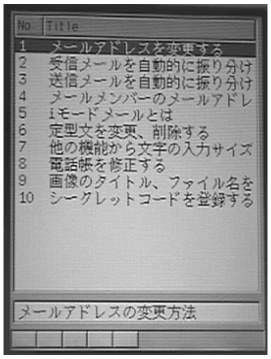
## 5. CELLPHONE USERS' MANUAL RETRIEVAL SYSTEM

In this section we describe a mike-input demonstration system where users speak into a microphone and then get system response on the screen. The mike-input system was implemented on the same evaluation board of the multicore processor as mentioned in the previous section. This system is a speech-input, text-output retrieval system which retrieves the passages of cellphone users' manual relevant to a spoken query sentence, and headlines the 10 best retrieval candidates. The spoken query sentence is recognized by the proposed parallel LVCSR module that is executed by the three CPU cores, and the recognition result is passed to the retrieval module that is executed by one CPU core. The retrieval module extracts key words from the recognition result, and search the users' manual for the 10 passages most relevant to the set of key words. The right-side photo in Fig. 3 shows the output screen of this system. The screen shows the 10 best candidates that are retrieved by the spoken query: "How can I change my e-mail address?". The recognition result is also shown on the bottom line.

The LVCSR module, retrieval module, and the cellphone users' manual are all implemented or stored on the evaluation board, and thus the system works without requiring any outside resources. The acoustic model is the same as that of the PDA translator though the language model is basically the same as that of the speech-input retrieval system over telephone [1], but is compactly reconstructed. The retrieval module is the same as that of the telephone system.

The system gives comfortable responses to spoken queries. It is not only much faster than the telephone system which uses both telephone connections and packet connections and has to wait for the connection closure and establishment between speech input and text output, but also requires no outside resources.





| No | Title |
|---|---|
| 1 | メールアドレスを変更する |
| 2 | 受信メールを自動的に振り分け |
| 3 | 送信メールを自動的に振り分け |
| 4 | メールメンバーのメールアドレ |
| 5 | iモードメールとは |
| 6 | 定型文を変更、削除する |
| 7 | 他の機能から文字の入力サイズ |
| 8 | 電話帳を修正する |
| 9 | 画像のタイトル、ファイル名を |
| 10 | シークレットコードを登録する |

メールアドレスの変更方法

Figure 3
Upper: Evaluation board
Right: 10 best candidates of the text retrieval

## 6. CONCLUSION

We proposed a parallel LVCSR algorithm for cellphone-oriented multicore processors. We introduced an acoustic look-ahead and blockwise computation to our compact LVCSR algorithm, in order to distribute its computational load to three CPU cores. We implemented the proposed algorithm on an evaluation board of a cellphone-oriented processor chip that has three ARM9 CPU cores with the operation frequency of 150MHz, and showed stand-alone LCVSR on cellphone within one real time can be achieved with recognition vocabulary of about 50,000 words. Using this parallel LVCSR, we also implemented on the same evaluation board a mike-input application system which was formerly implemented as a server-client system. The system searches for passages relevant to a user's spoken query, and then shows the 10 best candidates on the screen. The mike-input system provides comfortable responses to query sentences spoken through a microphone, which shows the possibility that in the near future users can get the usage of the cellphone or other information that is relevant to the the query sentence spoken to the cellphone itself, without requiring any outside resources.

## 7. REFERENCES

[1]S. Ishikawa et al., "Speech-activated Text Retrieval System for Multi-modal Cellular Phones", SP-P4.12, ICASSP, 2004

[2]R. Isotani et al., "An Automatic Speech Translation System on PDAs for Travel Conversation", pp.211-216, ICMI, 2002

[3]S. Torii et al., "A 600MIPS 120mW 70μA Leakage Triple-CPU Mobile Application Processor Chip", pp.136-137, ISSCC, 2005

[4]H. Ney et al., "A Word Graph Algorithm for Large Vocabulary, Continuous Speech Recognition", pp.1355-1358, ICSLP, 1994

[5]K. Shinoda et al., "Efficient Reduction of Gaussian Components Using MDL Criterion for HMM-Based Speech Recognition", pp.869-872, ICASSP, 2002

[6]T. Watanabe et al., "High Speed Speech Recognition Using Tree-Structured Probability Density Function", pp.556-559, ICASSP, 1995

[7]T. Hori et al, "A Study on a Phoneme-graph-based Hypothesis Restriction for Large Vocabulary Continuous Speech Recognition", Vol.40, No.4, IPSJ Journal (in Japanese), 1999