SCALABLE IMPLEMENTATION OF UNIT SELECTION BASED TEXT-TO-SPEECH SYSTEM FOR EMBEDDED SOLUTIONS

Nobuo Nukaga, Ryota Kamoshida, Kenji Nagamatsu and Yoshinori Kitahara

Hitachi, Ltd. Central Research Laboratory
1-280, Higashi-koigakubo Kokubunji-shi, Tokyo 185-8601 Japan
nukaga@crl.hitachi.co.jp

ABSTRACT

In this paper we propose two methods in order to implement unit selection-based text-to-speech engine into resource-limited embedded systems. While we have achieved improving the quality of synthesized speech by unit selection-based text-to-speech technology, there is a practical problem regarding the trade-off between the size of database and the quality of synthesized speech. That is, we need large database and expensive computation in order to generate highly natural sounding voices, and the text-to-speech system is required to meet the specification of target system. For this problem, we introduced frequency-based approaches to reduce the size of speech database. The experimental results showed the step-by-step downsizing method was better than the direct one in terms of the cumulative join cost and the target cost. Furthermore, some techniques were introduced and evaluated in order to implement our text-to-speech engine into an embedded system. From experimental results, it developed that the run-time work load for the test sentences was 80 MIPS approximately and the implemented engine was useful and scalable for mid-class embedded system.

1. INTRODUCTION

In recent years, many text-to-speech systems have been applied to various systems. Moreover, unit selection-based speech synthesizers [1][2] have been successful in improving quality of synthetic voices.

However, in order to synthesize highly natural sounding speech unit selection-based systems need large hardware resources, for instance, a gigabyte-order storage and random access memory and high performance CPU. Although there are few difficulties with the server solutions, the resources mentioned above does not meet the specification for many embedded systems, for example, home appliances, car navigation systems, personal digital assistance, cell phones, and so on. Moreover, text-to-speech engine is also required the lower computation in order not to interfere the other tasks because they are multi-processed simultaneously on the limited performance of RISC processor and real-time operating system. In general, the hardware requirement of the target system is determined prior to that of text-to-speech engine on the basis of hardware cost. Therefore, text-to-speech engine is required to change the size of resources on demand. This scalability is very important to implement into embedded system.

To sum up, we have two problems on this matter. One is that we need to downsize the offline database size so as to store it within the requirement, and the other is to reduce the online computation to select units and synthesize speech.

For these problems, some remarkable researches have been proposed. We classified the proposed methods into two groups in terms of approach. One is the method of reducing the size of database based on clustering or decision tree [3]. These approaches presume that candidates can be substituted for its similar one and the space of the candidates can be covered with the fewer ones. Using this approach, however, it might be occurred that the best segments for synthesis are omitted and the average ones are remained as the result of clustering. That is, it is likely that the quality of synthesized speech is indecisive and is lacking vividness. On the other hand, some methods we call frequency-based approach have been proposed. For instance, in unit selection and fusion method [4], the target waveforms are generated on the criterion of the frequency of the candidates which text-to-speech system selected actually as the result of reading test sentences. Most segments with high frequency are remained on frequency-based approach, so that the synthesized speech quality is expected to be equivalent to full size ones. In [4], however, the wave segments are derived by averaging the N-best candidates, therefore, it is likely that the quality is monotonous.

In this paper, we present two downsizing approaches to minimize the degradation, and we evaluate them in terms of the costs in section 3. For the second problem, a method of reducing the computational load and experimental results are discussed in section 4.

2. OUTLINE OF OUR JAPANESE TEXT-TO-SPEECH SYSTEM

2.1. Flow of TTS system

Fig. 1 shows the flow of our text-to-speech system. The input to our system is unlimited Japanese Kanji-Kana texts. Linguistic analysis module generates pronunciation symbol string from the input text. A pronunciation symbol string consists of phonetic alphabets and prosodic symbols which indicate the position of accent, pause and accentual phrase boundary. The process of the linguistic analysis is divided into morphological analysis, phrase dependency analysis, and phrasing and accentuation determination. Next, f0 contour, duration of phonemes and unit symbols are generated from the pronunciation symbols. Two prosodic parameters, f0 contour and duration of phoneme, are used for calculating target score. We use a super-positional model for calculating f0 contour and a context-dependent statistical model for calculating duration of phoneme. In the next unit selection step, the optimal speech segments are selected from speech database. Finally, the waveforms selected are concatenated and adjusted to the prosodic targets by signal processing technique. In our current system, we use TD-PSOLA [5] to put on the prosodic targets for voiced wave because of the low computational cost. Pitch waves are deleted and duplicated in order to fit the duration of the waveform to the target appropriately. As for unvoiced waves, the waveforms are copied with no modification.

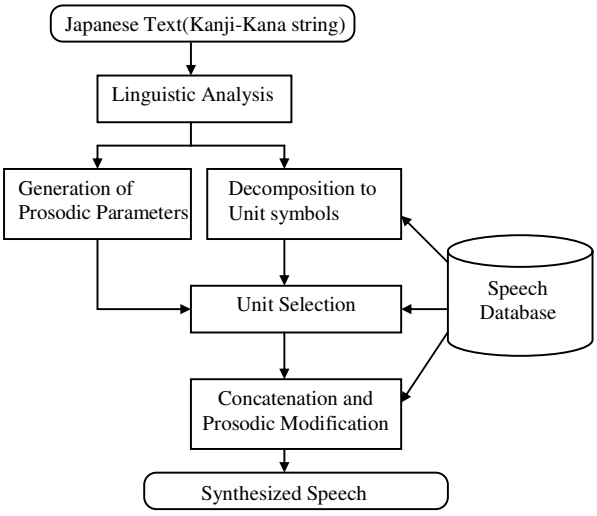

Fig. 1: Flow of text-to-speech system

2.2. Cost function and Unit Selection

Our cost function consists of join (concatenation) cost and target cost. The join cost is to evaluate the gap between two segments to be concatenated from the cepstral distance, F0, and pitch-synchronous cross correlation [6]. The join cost is normalized for all possible concatenation. The target cost is to evaluate between the candidates and the target. We use F0 and phoneme duration as the target. We do not calculate spectral target. In our system, the candidates are pre-screened by target cost first, and the optimal sets are determined from the pre-screened candidates using Viterbi search to minimize the sum of the cost of each candidates.

2.3. Building Database

Table 1 shows the summary of our initial speech database. It is the full size corpus for our TTS system. 796 sentences were spoken by a female professional speaker with normal speed and Tokyo Japanese accent and recorded at the rate of 22 kHz. Although we did not consider the phoneme balance on designing the script, we took the coverage into account in a way that the database included at least one candidate. The total record length of the waves was approximately 80 minutes. Consequently, the size of wave data was 207MB.

We divided the wave data into the segments based on the definition of unit. The unit consists of V_{fh}, V_{sh}, CV and VV. V_{fh} and V_{sh} mean the first half and the second half of vowel respectively. CV consists of the whole of consonant and the first half of vowel. VV is diphone, which consists of the second half of a vowel and the first half of the next vowel to handle diphthong. For example, a Japanese word *oboe* is decomposed into o_{fh}, o_{sh}, *bo*, *oe* and e_{sh}. Finally, 520 different units and 55025 segments were obtained.

Table 1: Summary of speech database

Number of sentences	796 sentences
Speaker specification	Professional, female and Tokyo Japanese accent
Speech rate	Normal
Sampling rate	22kHz
Recorded time	Approximately 80 minutes
Size of wave data	207MB
Unit	V_{fh}, V_{sh}, CV and VV
Number of different units	520
Number of segments	55025
Feature vector	cepstrum(14-th), Δcepstrum(14-th), power, Δpower and F0

3. PRE-SELECTION OF SPEECH DATABASE

3.1. Approach for Pre-selection of Speech Database

We introduce a method for reducing the number of candidates based on the cumulative frequency for each segment, in order to downsize speech corpus. The aim of adopting this pre-selection method is to maximize the coverage of the segments which are used in actual TTS system based on the assumption that we could maintain the overall speech quality if the frequently used segments would be remained. The target database is derived from the full size one automatically, whenever the requirement of target system is arisen.

In this approach, we have two choice of pre-selection. One is to reduce the size directly. For example, if the requirement is X megabytes which is equal to 30 percent of the full database, we pre-select the segments on which the cumulative frequency for N-best ones is 70 percent for each units. This approach is called direct reduction (DR). The other choice is to cut it down step-by-step. In this choice, the database is reduced by iterating direct reduction in a way that the final data size is within the specification. We call it step-by-step reduction (SR). Fig. 2 shows the illustration of the two presented methods.

In the next section, we will conduct an experiment to compare these approaches and to determine one in order to maintain better speech quality.

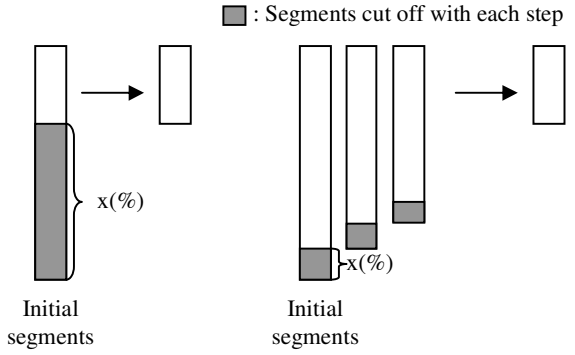

Fig. 2: Illustration of two methods for reducing database size. (Left: direct reduction, right: step-by-step reduction)

3.2. Experimental Results

In this section, we conducted an experiment to test the effectiveness of the methods described in section 3.1 by evaluating mean of the cumulative join cost for each phrase and the target cost for each segment. We assume that the cumulative join cost and the target cost has a positive relation to speech quality. In this experiment, we synthesized 50,269 Japanese sentences and recorded the costs for each sentence.

The results of the experiment are presented in Fig. 3 and 4. The dotted line shows the mean of the cumulative join cost in case of direct reduction and the solid line shows in case of step-by-step reduction, referred to as DR and SR respectively. The vertical axis means the mean of the cumulative join cost in Fig. 3 and the mean of the target cost in Fig. 4. These costs are normalized by our basis. The cumulative join cost was calculated for each phrase separated by punctuation and short pause, which means that the smaller, the better the quality. The horizontal axis means the size of wave data.

We set the final target of the database size set at under 10 MB for this experiment. In the method of SR, 10 percent of the segments were cut off in each step. Consequently, the database of approximately 9.4 MB was obtained by the iteration of 13 times. For the method of DR, the database of approximately 7.4 MB was obtained by cutting off directly 80 percent of the segments. From 10 to 70 percent of the segments were cut off for comparison. Each dot in these figures indicates the evaluated database.

From Fig. 3, SR was better than DR at the identical size of the speech database in terms of the mean of the cumulative join cost and that of the target cost. We assume that the join cost and the target cost is one of measures to determine the synthesized voice quality to which it has positive relation. From these results, we conclude that SR is better than DR to maintain the quality relatively in either case.

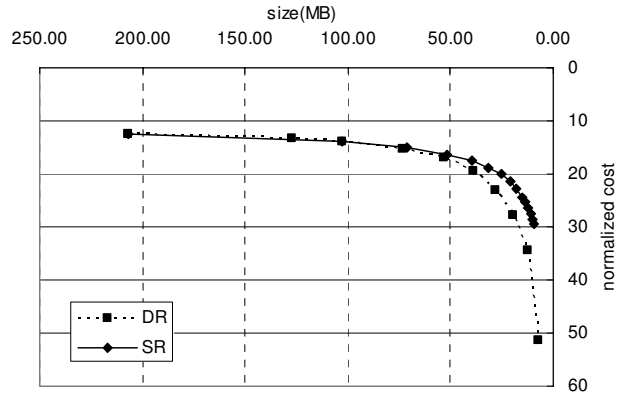

Fig. 3: Mean of the cumulative join cost for each phrase

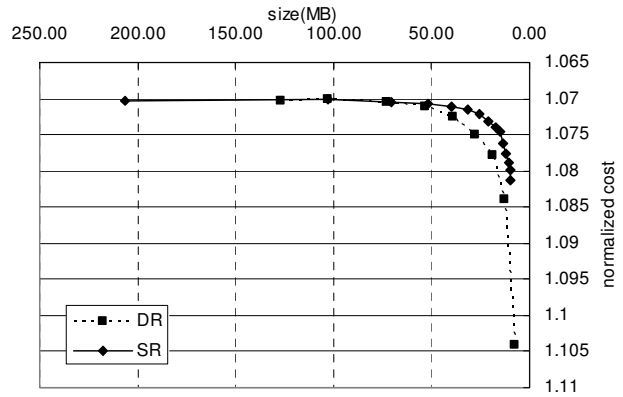

Fig. 4: Mean of the target cost for each segment

4. IMPLEMENTATION FOR TARGET SYSTEM

4.1. Target System for Evaluation

We adopted T-Engine tool kit released from Hitachi ULSI Systems in this evaluation. The T-Engine tool kit consists of T-Kernel upgraded from real-time µITRON operating system, SuperH™ RISC processor SH7751R(240MHz, 430MIPS), 64MB of DRAM, LCD and USB interface in

order to link external storage such as HDD, DVD, and so on. This tool kit enables us to develop prototypes for embedded system easily in terms of cost, availability and peripheral tools. The exterior of the target system for evaluation is shown in Fig. 5. T-Engine tool kit is shown on the left, and HDD which stores the program and the database of the text-to-speech system is shown on the right side.

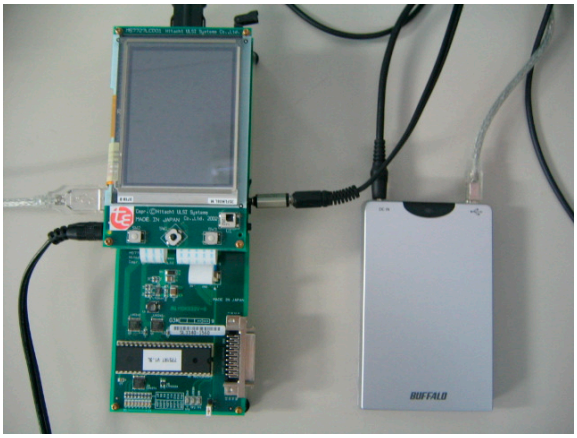

Fig. 5: Target system for evaluation

4.2. Offline Cost Calculation

During Viterbi search, it needs the computational power to calculate join costs for each pair of two successive candidates. We calculated the join cost for the possible combination of units by offline and stored with data files as the cost was read on demand.

4.3. Evaluation of Throughput

We evaluated the calculation time to synthesize from many input strings. 500 pronunciation symbol strings were used for this experiment changing overall data size. The overall data included speech waves, pre-calculated join cost and other parameter files. We evaluated E/L ratio in this experiment. E/L ratio was calculated by dividing E by L. E and L mean the execution time and the length of synthesized wave respectively on the condition that it takes E seconds to synthesize the speech waves with length of L seconds. This value can approximate the CPU load per unit time. For example, if the E/L ratio is 0.5, it means that it takes 500 msec to generate the waves of one second, in other words, the synthesizer needs to consume the half of CPU load.

Fig. 6 shows the average E/L ratio on the database size. This result shows that the synthesizer consumed 80 MIPS approximately on the version of 25 MB, which was obtained by multiplying 0.195 by the clock speed of 430 MIPS for SH7751R. This consumption of 80 MIPS is acceptable for mid-class embedded system such as car navigation system. Consequently, it was confirmed that the version of 25 MB or under was useful for these solutions.

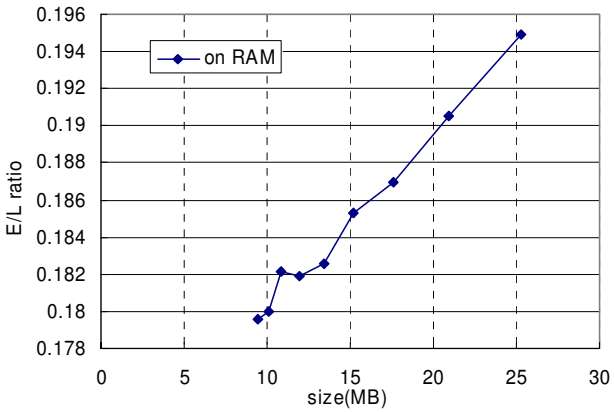

Fig. 6: E/L ratio on the database size

5. SUMMARY AND DISCUSSION

In this paper we presented the methods to downsize speech database and to implement a Japanese text-to-speech engine in order to meet the specifications for general embedded systems. To reduce the size of speech corpus, we introduced two methods in order to cut down the size of database and we evaluated by the sum of join costs and target cost.

From the experiment, it was proved that step-by-step reduction was better than direct one on the cumulative join cost and the target cost to maintain the speech quality. To implement Japanese text-to-speech engine, we introduced the offline cost calculation. Using these developments, our fully unit selection-based TTS system was implemented into T-Engine tool kit with SuperH™ RISC processor (SH7751R) and 64MB of RAM. From the experimental results, the runtime load for the tested sentences was approximately 80 MIPS. It developed that the implemented system was useful for mid-range embedded solutions.

REFERENCES

[1] A.J.Hunt and A.W.Black, "Unit selection in a concatenative speech synthesis system using a large speech database", Proc. of ICASSP'96, pp.373-376, 1996

[2] W.N.Campbell, "CHATR: A high-definition speech re-sequencing system," Proc. 3rd ASA/ASJ Joint Meeting, pp.1223-1228, 1996

[3] R.E.Donovan, "Segment Pre-selection in decision-tree based speech synthesis systems," Proc. of ICASSP 2000, pp.937-940, 2000

[4] M.Tamura, T.Mizutani and T.Kagoshima, "Scalable concatenative speech synthesis based on the plural unit selection and fusion method," Proc. of ICASSP 2005, pp.361-364, 2005

[5] E.Moulines and F.Charpentier, "Pitch synchronous waveform processing techniques for text-to-speech synthesis using diphones," Speech Communication, vol.9, pp.453-467, 1990

[6] N.Nukaga, R.Kamoshida and K.Nagamatsu, “Unit selection using pitch synchronous cross correlation for Japanese concatenative speech synthesis,” Proc. of ISCA 5th Speech Synthesis Workshop, pp.43-48, 2004