TEXT TO SPEECH SYNTHESIS SYSTEM FOR MOBILE APPLICATIONS

K. Partha Sarathy, A.G.Ramakrishnan*

Department of Electrical Engineering, Indian Institute of Science, Bangalore, India
parthu143@gmail.com, *ramkiag@ee.iisc.ernet.in

ABSTRACT

This paper discusses a Text-To-Speech (TTS) synthesis system embedded in a mobile. The TTS system used is unit selection based concatenative speech synthesizer, where a speech unit is selected from the database based on its phonetic and prosodic context. Speech unit considered in the synthesis is larger than a phone, diphone and syllable. Usually the unit is a word or a phrase. While the quality of the synthesized speech has improved significantly by using corpus-based TTS technology, there is a practical problem regarding the trade-off between database size and quality of synthetic speech, especially in mobile environment. Several speech compression schemes currently used in mobiles today are applied on the database. Speech is synthesized from the input text, using compressed speech in the database, The intelligibility and naturalness of the synthesized speech are studied. Mobiles contain a speech codec, one of the modules in the baseband processing. The idea of this paper is to propose a methodology to use the already available speech codec in the mobile and read a SMS aloud to the listener, when TTS is embedded in a mobile. Experimental results show the clear possibility of our idea.

Index Terms— corpus based concatenative TTS, RPE-LTP, CELP, ACELP, Automated book reader

1. INTRODUCTION

State of the art corpus-based text-to-speech (TTS) engines generate speech by concatenating speech segments selected from a large database. In the past few decades, TTS systems have been developed for desktop systems and other platforms, where enough hardware resources are available. Efforts are also made to build a TTS engine for embedded applications [1]. Corpus based TTS systems require more hardware resources for storing the large amount of recorded data. Synthesized speech quality is better if more data is available during training phase. However, TTS integrated in a mobile has limitations on processing and memory. There is a trade-off between the amount of recorded data to be used and the quality of synthetic speech in mobile environment.

Our idea is, given the limited number of resources available in the mobile environment, TTS should fit in it without interfering with the other modules. TTS in a mobile can be treated as a Value Added Service (VAS) for the user and usually given the least preference in terms of processing. This engine is not used periodically and hence need not continuously wait for the input in a mobile. As and when a SMS comes, user can enable TTS to speak out the text in the message. Care should be taken that the other functionalities in the mobile are not disturbed because of TTS integration. We have applied some of the commercially used speech codecs in mobiles on the database and found that TTS output quality is intelligible and natural. It is worth mentioning about flite[11] developed by CMU for research purpose. is used for research purpose. We are currently working on integrating our TTS in a commercial GSM mobile, which needs to be tested in real time.

2. OVERVIEW OF OUR TAMIL TTS SYSTEM

We have used TTS for Tamil, a south-Indian language for experimentation. A user or an Optical Character Recognition (OCR) system can give the input text to the TTS engine. TTS involves two modules. One is Natural Language Processing module, NLP [2] and the other is Signal-processing module. NLP module involves several tasks, which are related to a diverse set of disciplines . The NLP module takes input in the form of text and outputs a symbolic linguistic representation to the Signal Processing module. The latter outputs the synthesized speech based on its phonetic and prosodic context. The basic unit for speech synthesis is usually a phone, half-phone, diphone, syllable, word, sentence or any such unit. The naturalness of synthesized speech depends on the unit selection criteria [3]. These criteria are defined in terms of two costs namely,

- **Target cost,** which estimates the difference between the specifications of the database unit and the predicted input unit, based on good phonetic and prosodic models.

§ **Join cost or Concatenation cost,** which estimates the acoustic discontinuity between two successive joining units

The unit sequence selected from the database is the one with the lowest overall cost [4]. In our TTS engine, the basic unit is a word. The selection criteria uses only concatenation cost, since target cost calculation requires a prosody model, which we have not yet developed.

2.1 Database

The database used for our synthesis contains 1027 phonetically rich Tamil sentences with 5149 unique words. The coverage of the units is really good. The size of these wave files is about 295 MB on the hard disk. These sentences are recorded at 16 KHz sampling rate from a professional Tamil male speaker and the corresponding wave files have been segmented manually using PRAAT[10], a speech analysis program. Some sample wave files synthesized using the above database are available on the Medical Intelligence and Language Engineering Lab (MILE) website in IISc at the URL http://ragashri.ee.iisc.ernet.in/MILE/index_files/research%20area.html. As we increase the number of phonetically rich sentences in the database, the database size becomes larger and renders the TTS system impractical for embedded applications.

2. SPEECH COMPRESSION SCHEMES

The idea of this work is to use the speech codec already available in the mobile for compressing the above database. Speech codecs used in today's mobiles produce communication speech quality at low decoding complexity. The scheme used in mobiles is lossy compression, and hence the decompressed speech will not be exactly the same as the original. However, this speech is clearly perceived and hence is defined as communications speech quality. In mobile communications, speech that reaches a mobile is compressed and the decoder in the mobile decompresses and plays out on the speaker. Compressed speech is transmitted over the air interface to reduce the bandwidth of the signal. Human ear cannot perceive compressed speech and hence the speech should be decoded for understanding. It means that speech codecs are readily available in the mobiles. We want to use these speech codecs for synthesizing speech from text in a SMS. Today, in India, more people are using GSM mobiles and hence our research is concentrated mostly on GSM codecs. A GSM mobile uses GSM-FR codec [7], which uses RPE-LTP compression scheme. GSM-FR compresses 20 ms speech frame sampled at 16 KHz (320, 16-bit samples) to 260 bits with a bit rate of 13 Kbps. FR stands for Full Rate. The other low bit-rate commercial speech codecs used in GSM mobile are GSM-EFR AMR12.2, AMR10.2, AMR7.95, AMR7.4, AMR6.7, AMR5.9, AMR5.15 and AMR4.75 with bit rates of 12.2, 12.2, 10.2, 7.95, 7.4, 6.7, 5.9, 5.15 and 4.75 Kbps, respectively. EFR and AMR stand for Enhanced Full Rate [5] and Adaptive Multi Rate [6], respectively. GSM-EFR and GSM-AMR codecs are based on code-excited linear prediction (CELP) and Algebraic CELP, respectively. All the above-mentioned codecs are used in our experiments.

3. SPEECH SYNTHESIS

Input to our TTS engine can be direct text from a SMS. In our synthesis, units are searched based on their left and right contexts in the database. For any unit, many instances may be present in the database. If there are 'n' units in the input text and m_1, m_2..m_n instances of each unit in the database, then m_1 X m_2 X m_3 ...m_{n-1} X m_n combinations are possible. Join cost is calculated between instances of unit1 with those of unit2, instances of unit2 with those of unit3 and so on. One of the best paths is selected using viterbi search based on the lowest total join cost. The final set of units obtained after viterbi search are joined at appropriate positions, such that there is less mismatch at the points of concatenation. Optimum coupling [8] is used for coupling two successive units. The mismatch between two frames (last frame of the first unit and first frame of second unit) is taken as the Euclidean distance measure of 13 Mel-scale cepstral coefficients.

3.1 Database compression using speech codecs

We applied the compression schemes mentioned in section 2 on the database of 1027 wavefiles. The sizes of the database after different compression schemes are listed in Table 1.

Table 1 Database size for various compression schemes

Compression Scheme Applied	Database size for f_s = 16 KHz	Database size for f_s = 8 KHz
No compression	295 Mbytes	152 Mbytes
FR 13	36 Mbytes	21 Mbytes
EFR 12.2	35 Mbytes	20 Mbytes
AMR 12.2	35 Mbytes	20 Mbytes
AMR 10.2	30 Mbytes	18 Mbytes
AMR 7.95	24 Mbytes	16 Mbytes
AMR 7.4	24 Mbytes	15 Mbytes
AMR 6.7	22 Mbytes	14 Mbytes
AMR 5.9	20 Mbytes	13 Mbytes
AMR 5.15	18 Mbytes	12 Mbytes
AMR 4.75	17 Mbytes	12 Mbytes

3.2 Speech synthesis using compressed Database

The procedure for text to speech synthesis remains the same as discussed in section 1, with the exception that the units selected from the database are now compressed units. During the final stage of waveform generation, compressed speech units are decompressed, coupled and smoothened at concatenation points. It is then played out on the speaker. Exact compressed speech frame boundaries need to be considered during synthesis; even a single byte mismatch results in complete degradation of synthesized speech. This is because all the compression schemes use basic linear prediction principle, where correlation exists between frames and filter memories need to be updated continuously. To decode the initial frame of a compressed speech unit picked up from the database, the decoding algorithm resets the filter memories. Due to this the error gets propagated to all frames in the decoded speech unit. Actually, to decode this initial frame of compressed speech unit the decoder algorithm needs filter memories updated during the decoding of previous frame. Then, this results in error-free decoding of compressed speech unit. A block diagram of the synthesis using compressed database is shown in Fig. 1.

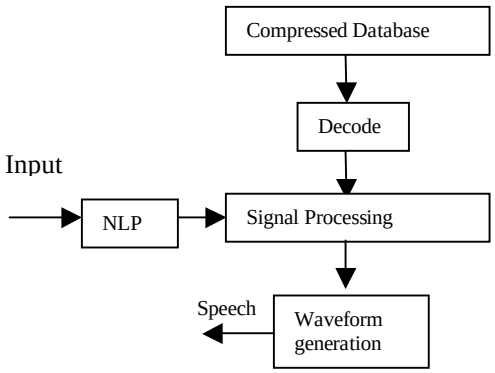

Figure 1 Block diagram of Text to speech in a mobile

4. PERCEPTION EXPERIMENTS

The performance of any TTS engine is usually evaluated based on the intelligibility and naturalness of the synthesized speech. We use mean opinion score (MOS) to assess the quality of our TTS output. Ten Tamil sentences, which are distinct from the 1027 sentences in the database, were synthesized. Wavefiles sampled at 16 and 8 Khz form the uncompressed database. The sentences are synthesized using uncompressed as well as compressed wavefile databases. Each of the 11 compression schemes listed in Table 1 is individually used for synthesis. Totally, 220 wavefiles are synthesized. Five native Tamil listeners were asked to rate these wave files for intelligibility and naturalness. MOS rating is defined as follows: 5 – Excellent 4 – Good 3 – Fair 2 – Poor 1 – Bad.

5. RESULTS AND DISCUSSION

The schemes used for experimentation are listed in Table 2. The mean of the ratings given by the five listeners are listed in Table 3 for each sentence.

Table 2 List of compression schemes used in our study.

Scheme A	16 Khz No compression
Scheme B	8 Khz No compression
Scheme C	16 Khz FR
Scheme D	8Khz FR
Scheme E	16Khz EFR
Scheme F	8Khz EFR
Scheme G	16Khz AMR 12.2
Scheme H	8Khz AMR 12.2
Scheme I	16Khz AMR 10.2
Scheme J	8Khz AMR 10.2
Scheme K	16Khz AMR 7.95
Scheme L	8Khz AMR 7.95
Scheme M	16Khz AMR 7.4
Scheme N	8Khz AMR 7.4
Scheme O	16Khz AMR 6.7
Scheme P	8Khz AMR 6.7
Scheme Q	16Khz AMR 5.9
Scheme R	8Khz AMR 5.9
Scheme S	16Khz AMR 5.15
Scheme T	8Khz AMR 5.15
Scheme U	16Khz AMR 4.75
Scheme V	8Khz AMR 4.75

Table 3. MOS ratings of the synthesized sentences. (mean of 5 listeners.)

No/ Scheme	A	B	C	D	E	F	G	H
1	3.3	4	4.3	4.7	4.3	4.3	3	3.7
2	3.3	4	4	4	4.3	4	3.7	3.7
3	4	4	4	4	4.3	4	4.3	4
4	3	3	3.3	3.3	3.3	3	3.3	3.3
5	4.7	4.7	4.7	4.7	4.7	4.7	4.7	4.7
6	4	4	3.7	4	4.3	4.7	4.7	4.7
7	3.7	3.7	3.7	3.7	3.7	3.7	3.7	3.7
8	4	3.7	3.7	3.7	4	4	4	4
9	4	4	4	4	4.3	4.3	4.3	4.3
10	3.3	3.3	3	3.3	3.7	3.7	3.7	3.7
Average	**3.7**	**3.8**	**3.8**	**3.9**	**4**	**4**	**3.9**	**4**

No/ Scheme	I	J	K	L	M	N	O	P
1	3	3	3	3.7	4	3.3	3.7	3.7
2	4	3.3	4	4.7	4	4	4	3.7
3	4	4.3	3.7	4.3	4	4	4	3.7
4	3.3	3	3.3	4	4	3.7	3.7	3.3
5	4.7	4.3	4.3	4.7	4.7	4.7	4.7	4.7
6	4.7	4.7	4.7	5	5	5	5	4.7
7	3.7	3.7	3.7	3.7	3.7	3.7	3.7	3.7
8	4.3	4.3	4.3	4.3	4.3	4.7	4.3	4.3
9	4.3	4.3	4.3	4.3	4.3	4.3	4.3	4.3
10	3.7	3.7	3.7	3.7	3.7	3.7	3.7	4
Average	**4**	**3.9**	**3.9**	**4.2**	**4.2**	**4.1**	**4.1**	**4**

No/ Scheme	Q	R	S	T	U	V
1	3.3	3.7	4.3	4	4	4
2	4	4	4.7	4	3.7	3.7
3	3.7	3.7	3.7	3.7	3.7	3.7
4	3.7	3.7	4	4	3.7	3.7
5	4.7	4.7	4.7	4.7	5	4.7
6	5	5	5	5	5	5
7	3.7	3.7	3.7	3.7	3.7	3.7
8	4	4	4	4	4	4.3
9	4.3	4.3	4.3	4.3	4.3	4.3
10	4	4	4	4	4	4
Average	**4**	**4.1**	**4.2**	**4.1**	**4.1**	**4.1**

The perception experiments evaluated that the TTS engine produced a high quality synthetic speech, even with highly compressed database. This, therefore, holds promise for the future, where we can read messages in Indian languages on the mobile. Good scores have been obtained for GSM FR and EFR, which means that optimized code can readily fit into GSM mobiles. This has a high potential for the market, since the synthesis quality is very good. However, AMR schemes are usually used when the communication channel is taken into consideration. We have used AMR in our experiments to understand the effect on synthesized speech quality at very high compression rates. Very high compression rates lead to very low memory requirement of the database. It's interesting to note that MOS score of synthesis using compressed data is very high compared to the case when uncompressed data is used. Listeners felt that the synthesized signal generated using compressed data has relatively smoother envelope. However, listeners also felt that pitch variation of units is locally good and needs some more modification in the global sense. Local refers to word level and global refers to a complete sentence or a phrase. Pause needs to be effectively modeled for improving the naturalness of the synthetic speech globally. The combination of high quality database and robust unit selection has resulted in good quality of our synthesis.

6. CONCLUSIONS AND FUTURE WORK

When a TTS is integrated in a mobile, ideally there are no constraints on the encoding process. However, the decoding complexity should be minimum and all commercial codecs used in mobiles satisfy this criteria. A new codec can be designed only for TTS applications in embedded devices [9]. Our intention is not to use additional memory in mobiles for a different codec meant only for TTS. Memory crunch always exists in embedded environments, especially in mobiles. The code size for TTS in our design is in the order of a few kilobytes. The challenge lies in selecting phonetically rich sentences in the database and fitting them into the mobile within the memory requirements and produce intelligible and natural speech. Efforts are in place to embed our TTS in a mobile and test it in real time. We are also progressing towards making a handheld embedded device called **'Automated Book reader for the Visually Challenged'** for Indian languages, which reads aloud pages from a printed book.

7. REFERENCES

[1] Nobuo Nukaga, Ryota Kamoshida, Kenji Nagamatsu and Yoshinori Kitahara. "Scalable Implementation of unit selection based text-to-speech system for embedded solutions", Hitachi Ltd. Central Research Laboratory, Japan.

[2] A. G. Ramakrishnan, Lakshmish N Kaushik, Laxmi Narayana. M, "Natural Language Processing for Tamil TTS", Proc. 3rd Language and Technology Conference, Poznan, Poland, October 5-7, 2007.

[3] A Black and N Campbell, "Optimizing selection of units from speech databases for concatenative synthesis", In *Proc, Eurospeech, pp.* 581-584, 1995.

[4] A Hunt and A Black, "Unit selection in a concatenative speech synthesis system using a large speech database", In *Proc. ICASSP*, pp. 373-376, 1996.

[5] Digital cellular telecommunications system (Phase 2+) (GSM); Enhanced Full Rate (EFR) speech transcoding (GSM 06.60 version 8.0.1 Release 1999).

[6] Digital cellular telecommunications system (Phase 2+) (GSM); Adaptive Multi-Rate (AMR); Speech processing functions; General description (GSM 06.71 version 7.0.2 Release 1998).

[7] Digital cellular telecommunications system (Phase 2); Full rate speech; Part2: transcoding (GSM 06.10 version 4.3.0 GSM Phase 2).

[8] S Isard and A D Coonkie. *Progress in Speech Synthesis*, chapter Optimum coupling of diphones. Wiley 2002.

[9] Chang-Heon Lee, Sung-Kyo Jung and Hong-Goo Kang "Applying a Speaker-Dependent Speech Compression Technique to Concatenative TTS synthesizers" *IEEE Trans Audio, Speech Lang. Proc., VoL. 15, No. 2, Feb 2007.*

[10] PRAAT : A tool for phonetic analyses and sound manipulations by Boersma and Weenink, 1992-2001. *www.praat.org*

[11] "Flite: a small, fast speech synthesis engine" Edition 1.3, for Flite version 1.3 by Alan W Black and Kevin A.Lenzo. *Speech Group at Cranegie Mellon University*