An Embedded and Concatenative Approach to TTS of Multiple Languages

Chen Gui-Lin, Han Ke-Song, Yu Zhen-Li, Yue Dong-Jian, Zu Yi-Qing

Motorola China Research Center, Motorola Labs

CITIC Square, 1168 Nanjing Rd. W., Shanghai, China

{a16825, a18186, a16300, a17498, a16534}@email.mot.com

Abstract

This paper presents an embedded and concatenative approach to multilingual text-to-speech system (ECMTTS). Under a uniform architecture, the TTS modules are separated into language dependent and independent ones. A specifically defined super phonetic symbol set enables to use uniform speech unit for concatenation, and an elaborately indexing and storing approach can reduce the size of speech inventory. The TTS system employs an improved cost function-based unit selection strategy, an efficient speech synthesizer, and refined concatenation approach to balance the speech quality and memory size as well as computation requirement on embedded platforms.

1. Introduction

The approach of waveform concatenation is still predominant in text-to-speech development. With recent progress of hardware, it becomes possible to use a large scale speech corpus to realize concatenation TTS with high quality and naturalness in many applications. The core idea of this approach is to select appropriate and variable length speech units from a big speech corpus based on minimizing acoustic distortions between selected units and targets [1]. Larger speech corpus provides more opportunities to get longer and suitable speech unit to minimize the number of splices and therefore discontinuities between contiguous units.

However, there is still a great amount of embedded TTS requests from hand held devices, such as mobile phone, PDA and automobile electronics. In these devices both computational power and memory resources are limited. Formant synthesis [2] or diphone based concatenation systems [3] may be the choices with their smaller footprint. But the quality and naturalness of these kinds of synthesis are not good enough in general.

In order to meet the requirement of handheld devices on the quality and complexity of TTS engine, the TTS system presented in this paper adopts concatenative synthesis technique with a compressed small footprint speech inventory. There are many challenges to build up an embedded and high quality TTS system, particularly in footprint constrains. It is impossible to introduce more speech unit with longer length, such as word, phrase and even utterance, to construct speech inventory. Because small speech units, such as phone, diphone, triphone or sub-word, comprise the speech inventory, there will be more spices within a concatenated utterance. A compromise cost function based concatenation seems un-successful in small speech inventory case. Using well defined sub-word speech unit and reusing parts of unit make it possible to keep natural sounding output with concatenation using small speech inventory.

In this paper, an approach is proposed to develop embedded and concatenative multi-lingual TTS systems. With this approach, natural quality of embedded TTS has been realized for English and 5 European languages: French (Fr.), Spanish (Es.), German (Gr.), Italian (It.) and Portuguese (Pt.). To efficiently use resources, the uniform methods and architecture are defined by referring the engineering realized, embedded English TTS system [5]. The modules of ECMTTS can be classified into two categories: language independent or common modules and language dependent modules. Unit selection and speech synthesis are language independent while text processing and speech data are language dependent. For the language dependent modules, the methods and architecture are uniform. For example, script design, text normalization, letter-to-sound (LTS) conversion, phonetic symbol expression, speech unit definition and data structure of speech inventory, etc., are all uniform.

In this paper, section 2 presents the system description, which includes description of the system architecture, the letter-to-sound conversion, the unit selector and the speech synthesizer. In section 3, the uniform definition of super set of phonetic symbol, the uniform definition of speech unit and speech inventory, and uniform strategy of speech unit selection and concatenation for the five European languages are described. The conclusion is in section 4, which discusses the quality of synthesis speech and language complexity.

2. System description

2.1 The architecture of TTS system

As shown in Figure 2-1, the uniform TTS system of European language is composed of three major modules: text processor, speech unit selector and speech synthesizer.

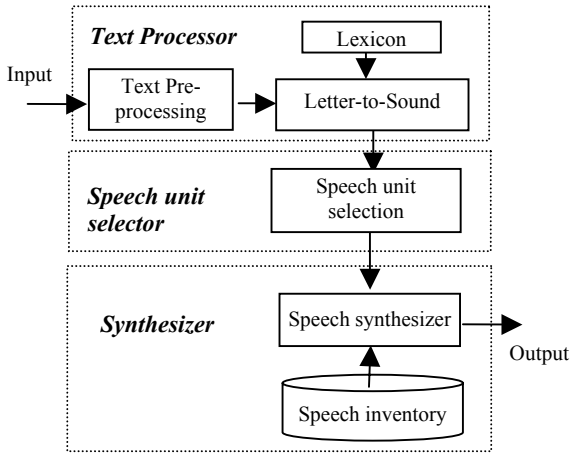

Figure 2-1 Architecture of embedded multilingual TTS

The text processor will conduct pre-processing for input text, such as text normalization for digit string, telephone number, time, date, currency, abbreviation and special symbols. Then the normalized text is converted into phonetic symbol string. According to the phonetic string and context, speech unit selector will select suitable speech units for concatenation. Finally, the speech synthesizer conducts speech unit concatenation on parameter level, and then decodes the parameter into output waveform.

2.2 Letter-to-sound conversion

Typically, there are two possible solutions to build a letter-to-sound module: 1) manually collect rules, with sufficient experience and language knowledge. This method is commonly referred to rule-based method; 2) acquire rules by machine learning, so called training-based method.

2.2.1 Rule-based approach

For a given language, if there exists a systematic relationship between a word format and its pronunciation, rule-based letter-to-sound can be efficient. Spanish, Italian, Portuguese and German are found as such kind of languages. A general rule can be described as:

$$[LS+]CC[+RS] \Rightarrow C_1C_2...C_i \rightarrow P_1P_2...P_j$$

The whole rule is read as: if the current character is CC, and the left and right content are LS and RS respectively, then the characters $C_1C_2...C_i$ are pronounced as phoneme string $P_1P_2...P_j$.

2.2.2 Training-based approach

Training-based method shows its advantages if there is no language knowledge available, or the work to write pronunciation rule set systematically is too difficult. Many comparative experiments show that it usually can achieve higher accuracy. For French, we use this method.

The training method contains three steps: 1) Letter-to-phoneme alignment; 2) Model training, including how to train the decision tree (CART tree is adopted in this study) and how to store the decision tree; 3) Pronunciation prediction.

2.3 Unit selector

The unit selection is based on cost function. Cost c_i of candidate phoneme p_i is defined as:

$$c_i = ul_i + ur_i + con_i \quad (1)$$

where, ul_i is left unit cost, ur_i is the right unit cost and con_i is concatenation cost.

Section 3 will give the details.

2.4 Speech synthesizer

Although concatenative speech synthesis has been proved to be a promising approach to practical TTS systems, the audible discontinuity at each concatenating point or between two contiguous units is an obstacle to speech quality. Here a method to realize the smooth transition at the concatenating point effectively is proposed, in which all the acoustic units are compressed with a customized CELP based coding scheme firstly. Then the unit concatenative synthesis is implemented. A detailed diagram is shown in Figure 2.4-1.

2.4.1 Compression of the speech inventory

The speech coding technique here is employed to compress the speech inventory of the concatenation based TTS system into a small footprint. Furthermore, the complexity of decoding operation should be low enough to be implemented on embedded system efficiently. By compromising the coding quality, bit-rate and complexity of decoding, a customized CELP based coding scheme is designed for the compression of speech inventory. All acoustic units of TTS inventory are encoded with a low bit rate and high quality CELP coder [6, 7, 8]. It reduces the memory requirement of the final TTS inventory significantly.

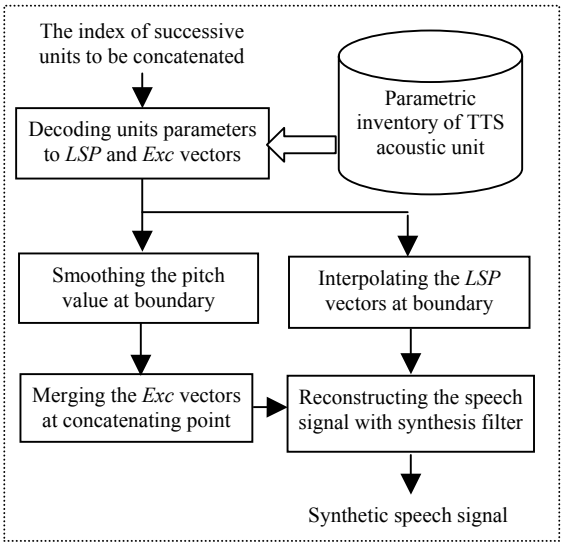

Figure 2.4-1.Unit compression and concatenative synthesis

2.4.2 Synthesis of unit concatenation

In concatenative synthesis, the bit-stream of units for synthesizing an utterance is firstly extracted from the speech inventory and decoded into vocal tract parameters *LSP* (Line Spectral Pairs) and excitation stream *Exc*. The pitch estimation of each frame is also obtained from the adaptive lag of the bit-stream.

The vocal tract response, excitation signal and pitch are interpolated and merged at concatenating point respectively. By filtering the merged excitation through synthesis filter, the smooth synthetic transition between two units may be generated.

3. Uniform methods for language dependent components

3.1 Super set of phonetic symbols for the languages

Defining a super set of phonetic symbol for multiple languages is helpful for uniform architecture. Based on phonological analysis, the number of phonetic identities of 5 European languages is shown in table 3.1-1. The super phonetic set, described by double-character symbol of phonetic alphabets in table 3.1-2, includes 65 phonetic identities (26 vowels, 30 consonants, and 9 semi-vowels and sonorant consonants). It should be pointed out that (1) the phonetic symbol set is designed by referring ARPAbet (used

for English) [9]; (2) same phonetic symbol may have slightly different acoustic correlation in different languages.

Table 3.1-1 Number of Phonetic identities

Language	Fr	Es	Gr	Pt	It
Phoneme number	42	34	47	42	44
Vowel number	18	5	23	14	16
Consonant number	16	20	18	19	19
Semi-vowel /nasal number	8	9	6	9	9

Table 3.1-2 Super phonetic symbol set for multiple languages

cv	Stop	*pp, pb,pf, bb, bz, tt, td, dd, dz, kk, kg, gg, gz*
	Fricative stop	*ff, vv, ss, sc, sh, hr, xh,*
	Fricative	*zz, zh, dh, ch, jh, th, ts*
sc	Label, nasal	*ll, lj, mm, nn, nj, ng*
	Vibrato	*rr, rd*
Sv		*ww, jj, jy,*
v	a-set	*aa, ae, an, am, au, ax, ay*
	e-set	*eh, en, ie,*
	i-set	*ih, ii, in, yi, ye, yh, yn, yo*
	o-set	*oa, on, ou, ow, oy*
	u-set	*uh, un, uw*

Where *c* is consonant; v is vowel; *nl* is nasal and labial; *sv* is semi-vowel; and *sc* stands for sonorant consonant.

3.2 Uniform definition of speech unit and inventory

Figure 3.1 is a universal syllable structure [10]. Among western languages, English may be the one of most complex languages in phonetic level. There are about 44 phonemes in English. Under the defined syllable structure, there are no constrains for phoneme combination. There are about 15 syllable patterns (see table 3.2-1). The rest of Western languages can share the same patterns. In fact, even tone language can share these syllable patterns. For example, Mandarin Chinese has simple syllable structure, which contains *c+v, c+v+nl, sv/sc/nl+v, and sv/sc/nl+v+nl.* The common definition of syllable structure is the basis of sharing uniform architecture and algorithms for multiple languages TTS.

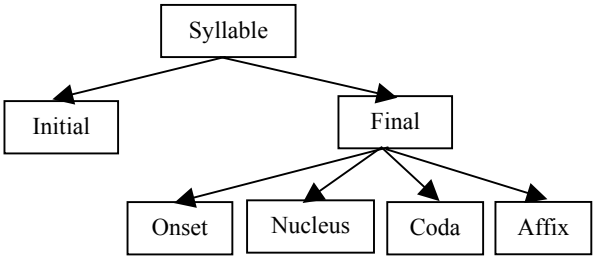

Figure 3.1 Syllable structure

For embedded concatenative multi-lingual TTS, the speech inventory of each language is the key component. Speech inventory should provide enough prosody featured speech units to meet the requirement of naturalness. Based on the syllable patterns listed in table 3.2-1, any pattern or sub-pattern can be used as speech unit. In other word, variable length sub-syllable constructs the speech inventory. Obviously, the smallest unit is phone.

On the other hand, the size limitation does not allow too many units to be included by speech inventory. Therefore, effective data structure of speech unit inventory incorporated with reasonable definition of cost function for unit selection is designed to compensate the insufficiency of speech inventory. According to table 3.2-1, there are 9 pairs of concatenation: *sv/sc+v, c+v, nl+sv, nl+v, v+sc/sv/nl, v+v, c+nl/sv/sc, v+c, c+c*.

Using Greedy algorithm and the information of unit frequency, a set of variable length speech unit is selected to compose speech inventory. There are phone (*c, v, sc, sv, nl*), di-phone (*sv+v, nl+v, c+v,* …), triphone (*c+v+sc/sv/nl, c+sv+v,* …), word and phrase in the speech inventory. All of speech units are extracted from a bigger speech corpus. This speech corpus has been labelled prosody information, including the boundary tags of prosodic word, prosodic phrase and intonation phrase. So that each speech unit has its prosody attributes.

Table 3.2-1 The patterns of syllable structure

Syllable structure	***Samples in English***
c+v (+c)	dog
c+v+sc/sv/nl(+c)	dark
c+sv+v (+c)	switch
c+sv+v+sc/sv/nl(+c)	screen
nl+sv+v (+c)	new
c+nl+v (+c)	sleep
c+nl+v +sc/sv/nl(+c)	clear
c+c+v(+c)	space
c+c+v+sc/sv/nl(+c)	spent
c+nl+v(+c)	smash
c+nl+v+sc/sv/nl(+c)	smart
c+c+nl+v(+c)	split
c+c+nl+v+sc/sv/nl (+c)	(ex)plain
c+sc+v(+c)	drive
c+sc+v+sc/sv/nl(+c)	drawn

3.3 Uniform strategy for unit selection and concatenation

Figure 3.3-1 shows the relationship between speech inventory, unit selector and speech synthesizer. Any part of speech unit can be selected to concatenate with other unit based on cost function [11, 12]. There are two concatenation types: brute-force concatenation and concatenation plus modification. Voiced-voiced concatenation always needs modification.

Here is an example to illustrate how to select variable length sub-unit based on concatenation cost: a speech unit has a *CVN* pattern in form of $C_iV_jN_k$. Here C_i is i^{th} consonant in a specific language, V_j is j^{th} vowel in that language, and N_k is k^{th} nasal. If i^{th} consonant and i'^{th} consonant belong to the same

articulation class, the cost function will point out the lower risk of the concatenation $C_{i\cdot}$ and V_jN_k. Therefore $C_iV_jN_k$ is a brute-force concatenated syllable. If C_iV_j in $C_iV_jN_k$. are used brute-force type to concatenate with $N_{k\cdot}$ to create $C_iV_jN_{k\cdot}$, the modification should be conducted because it is a voiced-voiced concatenation. During selecting speech unit, the prosodic information, such as the distance between current unit and left/right word boundary, break boundaries, are taken into account.

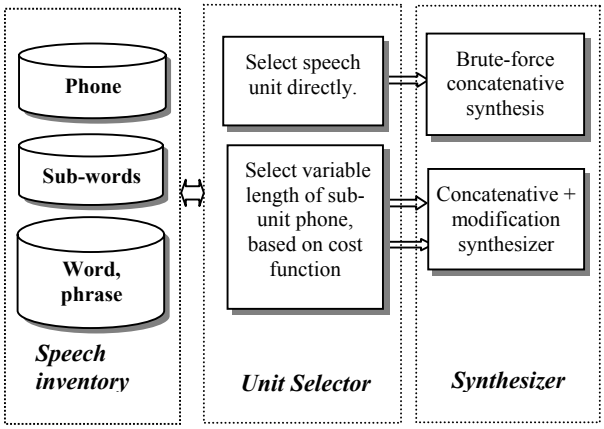

Figure 3.4-1 Speech inventory for embedded western language system

The brute-force concatenation is applied to the pairs: intra-syllabic *sv/sc/nl+c, c+v/sv/sc/nl*, *c+sv/nl/c,* and inter-syllabic *v+c, c+c.* Intra-syllabic *v+sv/sc/v/nl* and inter-syllabic *v+sv/nl/sc, sv/sc/nl/v* cases need modification when they are concatenated. The most difficult case for concatenation is intra-syllabic *sv/sc/nl+v,* where it has largest risk for concatenation and therefore more modification is needed.

4. Conclusions

In practice, it is very important to reach balance between footprint limitation and speech quality requirement in embedded TTS system. A uniform and variable-length unit at phonetic level for 5 European languages is defined to resolve this problem. Based on the definition of speech unit, the respectively designed text scripts and speech corpora for five languages have phonetic coverage under uniform structure. Consequently, the unit selection and concatenative synthesizer are common modules. Not only the common architecture has reduced complexity in developing TTS systems for multiple languages, but also the specialized method for speech concatenation and smoothing keeps the natural speech quality according to the underlying ideas. Many other languages, such as Danish and Swedish, demonstrate the similar phonetic features. Hence, the core ideas in this paper would be applicable to those languages, and even useful to meet challenges of language complexity in near future. Alternatively, the developed TTS systems show that the presented embedded approach can achieve higher naturalness than formant-based approaches in sense of prosodic variation and closeness to human speech.

From the point of view of language diversity, we do not think all components of a TTS engine can be trainable. Language dependent knowledge is important in text normalization, letter-to-sound conversion (whether training-based or other methods are adopted), speech inventory construction, prosodic controlling and system evaluation.

5. Reference

[1]. A.W. Black & N. Campbell, "Optimizing Selection of Unit from Speech Database for Concatenative Synthesis. " *Proc. Eurospeech95*.

[2]. Dennis H. Klatt, "Review of Text-to-Speech Conversion for English", *J.A.S.A.,* 82(3), pp. 737-791, 1987.

[3]. The MBROLA Project, Home page, 1999, Available on-line at [http://tcts.fpms.ac.be/synthesis/.

[4]. Jan van Santen, Richard, Sproat, *Introduction in Multilingual Text-to-Speech synthesis*, Kluwer Academic Publishers, 1998.

[5]. Guilin Chen, Dongjian Yue, Yiqing Zu, Zhenli Yu, "An Embedded English Synthesis Approach Based on Speech Concatenation and Smoothing", *ISCSLP2004*, Hong Kong, Dec, 2004

[6]. Dongjian Yue. "A Sub-band Speech Coding Scheme Based on Code Excited Linear Predictive Coding", *Proc. ISCSLP2000*, Beijing, China, 2000, pp. 113-116.

[7]. B. S. Atal, R. V. Cox, and P. Kroon, "Spectral quantization and interpolation for CELP coders," *Proc. ICASSP89*, Glasgow, U.K., 1989, pp. 69–72.

[8]. J. Adoul, P. Mabilleau, M. Delprat, and S. Morisette, "Fast CELP coding based on algebraic codes," *Proc. ICASSP87*, pp. 1957–1960.

[9]. J. Deller, J. Proakis, and J. Hansen, *Discrete-time processing of speech signals*, Macmillan, New York, 1993

[10]. A. Bell & J.B. Hooper, (eds). *Syllables and Segments*, North-Holland Publishing Company, 1978

[11]. Jon Rong-Wei Yi. *Natural-Sounding Speech Synthesis Using Variable-length Unit*. Master thesis, MIT, 1998.

[12]. Yiqing Zu, Guilin Chen, Zhenli Yu, Dongjian Yue "Concatenation Cost and Tonal Alignment", *From Traditional Phonology to Modern Speech Processing, Foreign,* Language Teaching and Research Press, Mar., 2004