Recent Enhancements in CU VOCAL for Chinese TTS-Enabled Applications

Helen Meng, Yuk-Chi Li*, Tien-Ying Fung*, Man-Cheuk Ho*,*
Chi-Kin Keung, Tin-Hang Lo*, Wai-Kit Lo* and P. C. Ching***

*Human-Computer Communications Laboratory,
Department of Systems Engineering and Engineering Management,
**Digital Signal Processing Laboratory,
Department of Electronic Engineering,
The Chinese University of Hong Kong,
Hong Kong SAR, China

{hmmeng, ycli, tyfung, mcho, ckkeung, thlo, wklo@se.cuhk.edu.hk, pcching@ee.cuhk.edu.hk}

Abstract

CU VOCAL is a Cantonese text-to-speech (TTS) engine. We use a syllable-based concatenative synthesis approach to generate intelligible and natural synthesized speech [1]. This paper describes several recent enhancements in CU VOCAL. First, we have augmented the syllable unit selection strategy with a positional feature. This feature specifies the relative location of a syllable in a sentence and serves to improve the quality of Cantonese tone realization. Second, we have developed the CU VOCAL SAPI engine, a version of the synthesizer that eases integration with applications using SAPI (Speech Application Programming Interface). We demonstrate the use of CU VOCAL SAPI in an electronic book (e-book) reader. Third, we have made an initial attempt to use the CU VOCAL SAPI engine in Web content authored with Speech Application Language Tags (SALT). The use of SALT tags can ease the task of invoking Cantonese TTS service on webpages.

1. Introduction

CU VOCAL[1] is a Cantonese TTS engine that we have developed using a syllable-based concatenative synthesis approach [1,2]. The unit selection strategy considers both coarticulatory context (in terms of distinctive features) of the left and right syllable neighbors as well as their tonal context. The approach is portable across Chinese dialects, e.g. from Cantonese and Putonghua.[2] The approach is also amenable to domain-optimization – a procedure that can be applied to improve the quality of synthesis for specific application domains, e.g. stocks, air travel, etc. Our current work aims to improve general tone realization of CU VOCAL by including a positional feature in the unit selection strategy that accounts for tone declination.

We have also been devoting significant development effort to ease integration of CU VOCAL with voice-enabled applications. Previously we have replaced the original distribution of dynamic linked libraries (DLL) and a home-grown application programming interface (API) with Web services based on Microsoft's .NET [3] platform to produce the CU VOCAL Web service [4]. CU VOCAL Web service is one of the first text-to-speech synthesis Web services. It is a modular software application that publishes the API over the Web with WSDL service description [5]. It allows other Web services or clients to invoke the CU VOCAL Web service at runtime. This facilitates development of scalable and extensible voice-enabled applications by integrating the CU VOCAL Web service with other Web services across different platforms and systems. For example, we have integrated the CU VOCAL Web service with a telephone alert Web service to form the CU VOCAL Voice Alert System [4]. The user can input his/her profile that includes a phone number (or multiple numbers for party calls and broadcasts), a dynamic information source and a related alert condition, e.g. a financial index rising above a specified level, the time of arrival of a flight, etc. When the alert condition is met, the CU VOCAL Voice Alert System will synthesize a spoken message based on the real-time information and deliver this message by calling the prescribed telephone number(s). Our current work seeks to develop a version of CU VOCAL with a standardized API to ease integration with applications. We focus on SAPI (Speech Application Programming Interface) [6] and will refer to this enhancement as the CU VOCAL SAPI engine. With this we can easily augment SAPI-compliant applications with Cantonese TTS capability, such as the electronic book (e-book) reader. We have also made an initial attempt to use the CU VOCAL SAPI engine to read Web content authored with Speech Application Language Tags (SALT) [6]. The use of SALT tags can ease the task of invoking Cantonese TTS service on webpages.

In the following we will describe our work on (1) improving tone realization of CU VOCAL; (2) developing the CU VOCAL SAPI engine and (3) developing SALT-enabled webpages that can invoke the CU VOCAL SAPI engine.

2. Improving Tone Realization in CU VOCAL

Tone realization is important for Cantonese TTS since the dialect has a very rich tonal structure, as illustrated in Figure 1. The acoustic realization of the syllable tone is the fundamental frequency (*f0*). *f0* is in turn affected by the tones of the neighboring syllables. CU VOCAL captures this effect during unit selection by taking the tonal context of the left and right neighboring syllables into consideration [9]. However this approach does not account for the declination effect on *f0*.

[1] URL: www.se.cuhk.edu.hk/cuvocal/

[2] Putonghua is the official Chinese dialect and Cantonese is a major Chinese dialect spoken in Hong Kong, South China and many overseas Chinese communities.

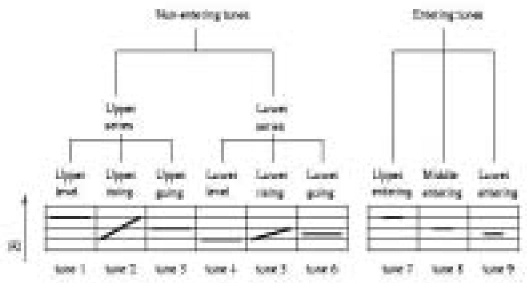


Figure 1. The Cantonese nine-tone system [8]. Non-entering tones are longer in duration while entering tones are shorter. Tones 1,3,4 and 6 have a flat tone shape and are distinguished mainly by their tone heights. The remaining tones have a rising tone shape.

Declination is the tendency of the f0 to decline over the course of an utterance [10,11,12].[3] Hence declination affects tone realization in Cantonese TTS. This is illustrated in Figure 2, which shows the pitch contour corresponding to 於天空飛的禿鷹深居山谷 (translation: "the hawk which is flying in the sky lives in the valley").[4] The utterance is pronounced as the tonal syllable sequence /jyu1 tin1 hung1 fei1 dik1 tuk1 jing1 sam1 geoi1 saan1 guk1/. To better demonstrate the tone declination effect, this utterance is carefully designed such that all syllables are in tone 1 (which are expected to have the same pitch level when they are spoken in isolation). The dotted lines indicate the drop in f0 as we move from the earlier syllables to the later.

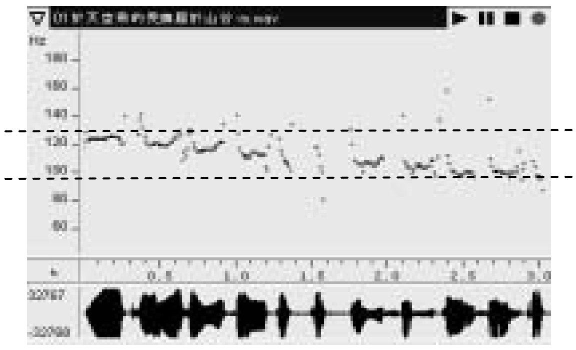


Figure 2. Illustration of the pitch contour corresponding to the recorded phrase 於天空飛的禿鷹深居山谷. All syllables in this utterance are in tone 1. We observe the declination effect by the drop of f0 across syllables.

The effect of declination seems perceptually more prominent for the "flat" tones (tone 1, 3, 4 and 6) when compared to rising tones (tones 2 and 5). For rising tones, the start and end points of the pitch contours need to maintain continuity with the left and right syllable tones and declination may thus have less of a perceptual influence. On the other hand, "flat" tones are mainly distinguished by their *f0* levels and these are directly affected by declination. For example, consider the Chinese character 新 (translation: "new") that is pronounced as /san1/ with tone 1 and a high *f0*. If we segment the syllable waveform of /san1/ from the acoustic wave of an utterance that contains 新 near the utterance final position and play it in isolation to a human subject, it is often mistakenly perceived as the character 腎 (translation: "kidney") that is pronounced as /san6/ with tone 6 (i.e. a lower *f0*). In Cantonese TTS, this will affect tone realization – for example, synthesis of the phrase [新] 的 功 能 (translation: a new function) that should be pronounced as /**san1** dik1 gung1 nang4/ will sound like the phrase [腎] 的 功 能 (translation: the kidney's function) that should be pronounced as /**san6** dik1 gung1 nang4/.

We modified the unit selection strategy in CU VOCAL in order to account for the declination effect. We augmented the feature set (that captures co-articulation and tone) with a *positional feature*. This feature specifies the relative location of a syllable in an utterance – START, NEAR-START, CENTER, NEAR-END, END. In a given utterance, START is the first syllable, END is the last syllable and the remaining features (NEAR-START, CENTER, NEAR-END) mark three positional groups sharing approximately equal proportions. The incorporation of the positional feature in unit selection enables CU VOCAL to minimize discrepancies in tone realization that are caused by declination. The improvement on both intelligibility and naturalness was tested to be statistically significant (α=0.01). The test involves twenty impartial subjects listening to ten pairs of wave files. The pair consists of synthesis outputs with and without using the positional feature and is played in randomized order. The positional feature is used only for declarative sentences and not for questions.[5]

3. CU VOCAL SAPI Engine

An earlier distribution of CU VOCAL is in the form of a dynamic linked library (DLL) with a home-grown API. As we begin to integrate CU VOCAL into various applications, the desirability of a standardized API across applications became compelling. For example, newspaper content in Hong Kong often contain mixed Chinese and English text, hence a newspaper reader needs to invoke both CU VOCAL and an English TTS engine. If different TTS engines have different APIs, integration with the application becomes cumbersome. (See Figure 3).

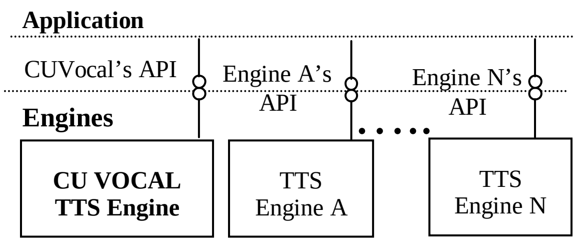


Figure 3. An application may need to invoke multiple TTS engines. If each has its own specific API, system integration becomes a cumbersome task.

We investigate the use of Microsoft's Speech Application Programming Interface (SAPI). SAPI is a Windows-based API for speech application development. SAPI-compliant speech engines are accessible through the component object model (COM) [14], a standard framework for developing and supporting program component objects.

[3] Declination applies with the exception of questions that have a rising intonation towards the end of the utterance.

[4] The pitch contour is obtained using KTH's *wavesurfer* [13].

[5] When CU VOCAL detects that the input text is a question, the positional feature is not used for synthesis. We are currently studying tone declination in utterances that are questions.

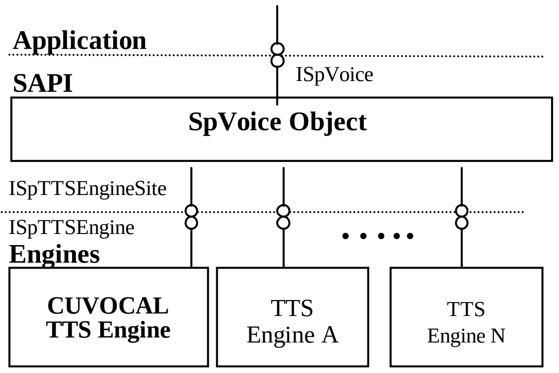


Figure 4. Integration of SAPI-compliant speech (TTS) engines in an application. The application only needs to invoke SAPI-related speech functions (such as IspVoice). SAPI, in turn, interfaces with individual TTS engines – ISpTTSEngineSite and ISpTTSEngine are interfaces for such communication.

A speech engine becomes a component object under the COM framework. Applications can invoke different SAPI-compliant speech engines by means of standardized SAPI calls. This is illustrated in Figure 4.

The CU VOCAL SAPI engine is developed on Microsoft's .NET platform. The sample engine provided by the SAPI SDK is used to wrap around the CU VOCAL core engine and to handle all calls to it. The MakeVoice package in SAPI is used to set the system's registry for the CU VOCAL SAPI engine in the machine. Figure 5 illustrates this integration process.

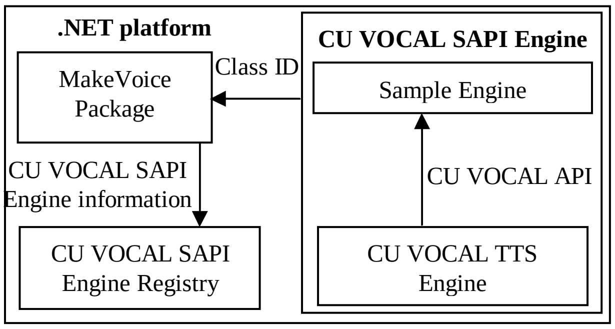


Figure 5. Using the .NET platform, the sample engine from SAPI SDK is used to wrap the CU VOCAL TTS core engine and handle all calls to it. The Class ID of the CU VOCAL SAPI Engine is passed to the MakeVoice package (provided in SAPI) to set the information of this SAPI engine in the system's registry.

3.1 Shared Memory

The development of the CU VOCAL SAPI engine facilitates ease of integration with applications that can invoke SAPI-compliant engines. It is possible for *multiple* Windows applications to invoke the CU VOCAL SAPI engine concurrently. Under the current SAPI specification, every instantiation of a SAPI-compliant engine spawns a new process for handling requests to the engine. For the CU VOCAL engine, multiple instantiations imply that computing resources will be tied up with duplicate copies of the pronunciation lexicons and syllable libraries. To solve this

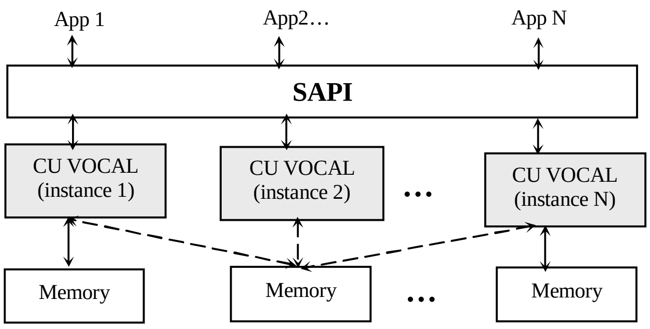
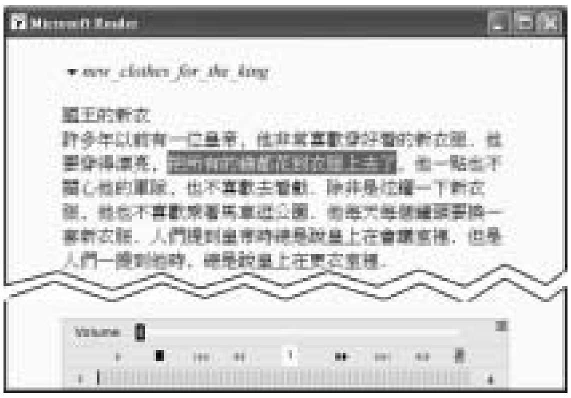


Figure 6. A shared memory implementation for the CU VOCAL SAPI engine. This supports concurrent use of the Cantonese TTS synthesizer by multiple Windows-based applications. The memory-mapped file avoids duplicate copies of pronunciation lexicons and syllable libraries.

problem of inefficiency, we have re-engineered CU VOCAL SAPI engine with a *memory-mapped file* to avoid duplicate copies of lexicons or libraries.[6] The memory resources are initialized only when the first instance of the CU VOCAL SAPI engine is instantiated. Any subsequent instances will share the originally initialized memory so that resources will not be used inefficiently. The tied resources are released after the termination of the last CU VOCAL instance (see Figure 6).

3.2 A Cantonese TTS Electronic Book Reader

We demonstrate the use of the CU VOCAL SAPI engine in the Microsoft Reader [15]. This is an electronic book (e-book) reader available in the Microsoft Windows platform that can read out English text by a SAPI-compatible English TTS synthesizer. We have integrated the CU VOCAL SAPI engine with the Microsoft Reader to enable it to read out Chinese textual content (with traditional Big5 encoding) in Cantonese. The Reader can highlight the textual phrase as it is being synthesized / read. A screen capture of the Cantonese e-book reader is shown in Figure 7.



new_clothes_for_the_king

國王的新衣

[illegible]

Figure 7. Screenshot of our Cantonese Electronic Book Reader developed by integrating the Microsoft Reader with the CU VOCAL SAPI engine. The Reader highlights the chunk of text as it is being read.

[6] Managing Memory-Mapped Files in Win 32 http://msdn.microsoft.com/library/en-us/dngenlib/html/msdn_manamemo.asp?frame=true

4. SALT-Enabled Cantonese TTS Webpages

We have made an initial attempt to use the CU VOCAL SAPI engine for Web content authored with Speech Application Language Tags (SALT). SALT is a speech interface markup language that extends existing Web markup languages to enable multi-modal and telephony access to Web content [7]. It consists of a small set of XML elements, with associated attributes and DOM (Document Object Model) [16] object properties, events and methods, which apply a speech interface to webpages. A SALT-enabled webpage can be authored with the use of SALT tags and read with a SALT-enabled browser. The SALT tags facilitate easy invocation of TTS service on the Web. We have integrated the CU VOCAL SAPI engine with SALT and illustrate its use with an example SALT-enabled webpage. This webpage (see Figure 8) can synthesize speech from prompt text (see red Chinese text in Figure 8) and Chinese text (in Big5 encoding) input via the text box. Figure 9 shows the underlying software implementation with SALT tags that invoke the CU VOCAL SAPI engine.

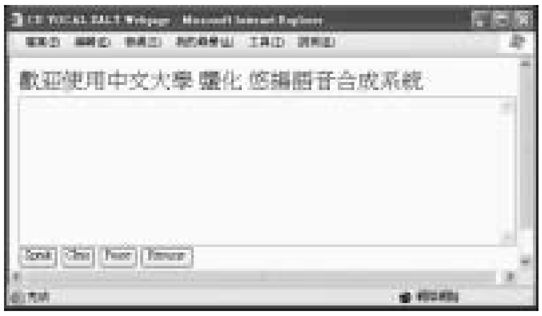
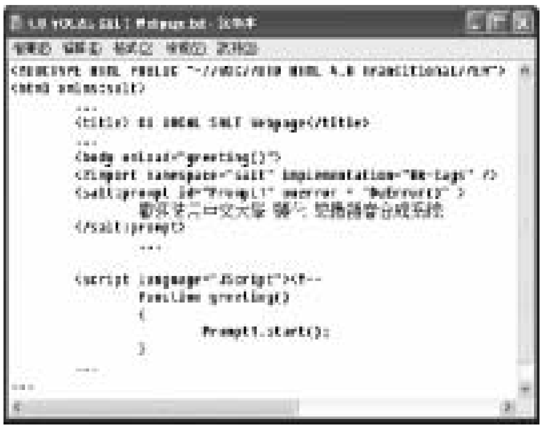

CU VOCAL SALT Webpage - Microsoft Internet Explorer
歡迎使用中文大學 [illegible] 語音合成系統
Speak
Clear
Pause
Resume

Figure 8. A SALT-enabled Cantonese TTS webpage



```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html xmlns:salt>
        ...
        <title> CU VOCAL SALT Webpage</title>
        ...
        <body onload="greeting()">
        <?import namespace="salt" implementation="MS-tags" />
        <salt:prompt id="Prompt1" onerror = "OnError()" >
                歡迎使用中文大學 [illegible] 語音合成系統
        </salt:prompt>
                ...

        <script language="JScript"><!--
                function greeting()
                {
                        Prompt1.start();
                }
        ...
...
```

Figure 9. Underlying software implementation for Figure 8 illustrating how SALT tags were used.

5. Conclusions and Future Directions

This paper reports on three recent enhancements to CU VOCAL, our Cantonese text-to-speech synthesizer. They include: (1) Incorporation of a positional feature in unit selection for syllable-based concatenative synthesis – the positional feature aims to improve tone realization in synthesis by accounting for declination effects. (2) Development of the CU VOCAL SAPI engine that eases integration with applications that can invoke SAPI – we have also implemented a shared memory feature for efficient use of computing resources when CU VOCAL is used by multiple applications concurrently. A demonstration system is provided by the Cantonese electronic book reader, developed by integrating the CU VOCAL SAPI engine with the Microsoft Reader. (3) Development of SALT-enabled Cantonese TTS webpages. The use of SALT tags can easily augment Web content with speech capabilities. Incorporation of the CU VOCAL SAPI engine in a SALT webpage facilitates the use of Cantonese TTS to read out Chinese Web content. Related applications will be particularly useful for the visually impaired, the elderly and uses in hands-busy/eyes-busy environments.

6. Acknowledgments

This work was partly funded by the Innovation and Technology Fund from the Hong Kong SAR Government (CUHK ITS/117/01) and also by a grant from Microsoft Research. We wish to thank Dr. Alex Acero and Dr. Kuansan Wang from Microsoft Research, and Dr. Eric Chang and Dr. Min Chu from Microsoft Research Asia for their help and comments on this work.

7. References

1. Fung, T. Y. and Meng, H. Concatenating Syllables for Response Generation in Domain-Specific Applications. Proc. of ICASSP, Istanbul, 2000.
2. Meng, H. et. al. CU VOCAL: Corpus-based Syllable Concatenation for Chinese Speech Synthesis across Domains and Dialects. Proc. of ICSLP, Denver, 2002.
3. Microsoft .NET : http://www.microsoft.com/net/
4. Meng. H. et. Al. CU VOCAL Web Service: A Text-to-speech Synthesis WebService for Voice-enabled Web-mediated Applications. Proc. of WWW, Budapest, 2003.
5. W3C Web Services Description Language (WSDL). http://www.w3.org/TR/wsdl
6. SAPI5.1 http://www.microsoft.com/speech
7. SALT forum http://www.saltforum.org
8. Lee, T., Meng, H., Lau W., Lo, W. K., and Ching, P. C., "Micro-prosodic Control in Cantonese Text-to-Speech Synthesis," in Proc. of the European Conference on Speech Communication and Technology, Vol.4, pp.1855 - 8, 1999.
9. Fung, T. Y. and Meng, H. The Effect of Tonal Context on Cantonese Concatenative Speech Synthesiss. Proc. of ISCSLP, Taipei, 2002.
10. Hart and Cohen "Intonationby rule: a perceptual quest", Journal of Phonetics, 1, pp. 309-327, 1973.
11. Hart et al. and Cohen, A.,"Declination: construct or intrinsic feature of speech pitch?", Phonetica, 39, pp.254-273, 1982.
12. Fujisaki, H. and Hirose, K., "Analysis of Voice Fundamental Frequency Contours for Declarative sentences of Japanese," in Journal of Acoustic Society of Japan (E) 5(4): 233-241, 1984.
13. Wavesurfer, http://www.speech.kth.se/wavesurfer/
14. Microsoft COM technology http://www.microsoft.com/com
15. Microsoft Reader, http://www.microsoft.com/reader/
16. W3C Document Object Model, http://www.w3.org/DOM